Package: gofCopula (via r-universe)

October 20, 2026

Type Package

Title Goodness-of-Fit Tests for Copulae

Version 0.4-3

Date 2025-08-26

LazyLoad yes

LazyData true

Depends R (>= 1.9.0), copula (>= 1.0-1)

Imports foreach, parallel, doSNOW, R.utils, SparseGrid, numDeriv, VineCopula (>= 2.0.5), methods, stats, MASS, utils, yarrr, progress, crayon

Description Several Goodness-of-Fit (GoF) tests for Copulae are provided. A new hybrid test, Zhang et al. (2016) <doi:10.1016/j.jeconom.2016.02.017> is implemented which supports all of the individual tests in the package, e.g. Genest et al. (2009) <doi:10.1016/j.insmatheco.2007.10.005>. Estimation methods for the margins are provided and 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. For reproducibility of results, the functions support the definition of seeds. Also all the tests support automatized parallelization of the bootstrapping tasks. The package provides an interface to perform new GoF tests by submitting the test statistic.

License GPL (>=3)

NeedsCompilation yes

RoxygenNote 7.3.2

Config/pak/sysreqs libgsl0-dev libicu-dev libjpeg-dev pari-gp

Repository https://simontrimborn.r-universe.dev

Date/Publication 2025-08-26 14:28:08 UTC

RemoteUrl https://github.com/simontrimborn/gofcopula

RemoteRef HEAD

RemoteSha d10455783bbdc78c84e93731efe8c413441de45e

Contents

Banks . . . 3
CopulaTestTable . . . 3
CryptoCurrencies . . . 4
gof . . . 4
gofArchmChisq . . . 7
gofArchmGamma . . . 10
gofArchmSnB . . . 14
gofArchmSnC . . . 17
gofCheckTime . . . 20
gofco . . . 22
gofCopula4Test . . . 25
gofCustomTest . . . 25
gofCvM . . . 28
gofGetHybrid . . . 31
gofKendallCvM . . . 32
gofKendallKS . . . 35
gofKernel . . . 38
gofKS . . . 42
gofOutputHybrid . . . 45
gofPIOSRn . . . 46
gofPIOSTn . . . 49
gofRosenblattChisq . . . 52
gofRosenblattGamma . . . 56
gofRosenblattSnB . . . 59
gofRosenblattSnC . . . 62
gofTest4Copula . . . 65
gofWhich . . . 66
gofWhichCopula . . . 66
gofWhite . . . 67
IndexReturns2D . . . 70
IndexReturns3D . . . 70
plot.gofCOP . . . 71
print.gofCOP . . . 72
print.goftime . . . 73

Index **74**

Banks — *Volatility-adjusted log returns of the two Banks Citigroup and Bank of America.*

Description

A dataset containing the volatility-adjusted log returns of two Banks in the years 2004-2012.

Format

A list with 9 entries containing the Banks log returns divided by years.

Source

Yahoo-Finance.

CopulaTestTable — *Applicable dimensions and copula for each test*

Description

`CopulaTestTable` returns a table which shows the applicable dimensions and copula for each test.

Usage

```
CopulaTestTable()
```

Details

Before performing a gof test with the package on a dataset, it pays out to know the implemented copula and dimensions for each test. This function is dedicated to help finding the applicable tests depending on the copula and dimensions available.

Value

A character matrix which consists of dimensions for the combination of tests and copula.

Examples

```
CopulaTestTable()
```

CryptoCurrencies *Volatility-adjusted log returns of the two Cryptocurrencies Bitcoin and Litecoin.*

Description

A dataset containing the volatility-adjusted log returns of two Cryptocurrencies in the years 2015-2018.

Format

A list with 4 entries containing the Cryptocurrencies log returns divided by years.

Source

CoinMetrics. https://coinmetrics.io/

gof *Combining function for tests*

Description

gof computes for a given dataset and based on the choices of the user different tests for different copulae. If copulae are given, all the implemented tests for those copulae are calculated. If tests are given, all the implemented copulae for every test are used. If both copulae and tests are given, all possible combinations are calculated.

Usage

```
gof(
  x,
  priority = "copula",
  copula = NULL,
  tests = NULL,
  customTests = NULL,
  param = 0.5,
  param.est = TRUE,
  df = 4,
  df.est = TRUE,
  margins = "ranks",
  flip = 0,
  M = 1000,
  MJ = 100,
  dispstr = "ex",
  m = 1,
```

```
  delta.J = 0.5,
  nodes.Integration = 12,
  lower = NULL,
  upper = NULL,
  seed.active = NULL,
  processes = 1
)
```

Arguments

x	A matrix containing the data with rows being observations and columns being variables.
priority	A character string which is either "tests" or "copula". "tests" indicates that all implemented tests are performed for all copulae which the tests share. These are e.g. "normal" and "clayton". If "copula" is chosen, all copula are chosen and only the tests are performend which these copula share. If one of the arguments tests or copula is not NULL, then priority doesn't affect the choice of the copulae and tests.
copula	A character vector which indicates the copula to test for. Possible are "normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos", "huslerReiss", "tawn", "tev", "fgm" and "plackett".
tests	A character vector which indicates the tests to use. Possible choices are the individual tests implemented in this package.
customTests	A character vector which indicates the customized test to use, if any. The test has to be loaded into the workspace. Currently the function containing the test has to have 2 arguments, the first one for the dataset and the second one for the copula to test for. The arguments have to be named "x" and "copula" respectively.
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. For the "gofPIOSTn" test the entry is limited to 60 degrees of freedom for computational reasons.
df.est	Indicates if df shall be estimated. Has to be either FALSE or TRUE, where TRUE means that it will be estimated. For the "gofPIOSTn" test the estimate is limited to 60 degrees of freedom for computational reasons.
margins	Specifies which estimation method for the margins shall be used. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively the following distributions can be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp"). Input can be either one method, e.g. "ranks", which will be used for estimation of all data sequences. Also an individual method for each margin can be specified, e.g. c("ranks", "norm", "t") for 3 data sequences. If one does not want to estimate the margins, set it to NULL.
flip	The vector of control parameters to flip the copula by 90, 180, 270 degrees clockwise. Only applicable for bivariate copula. Default is 0 and possible inputs

are 0, 90, 180, 270 and NULL. One can either specify one flip degree which will be applied on all copulae or choose an individual flip for each copula in which case the input has to be a vector.

`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.

`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. Only necessary if the test `gofPIOSTn` is part of `tests`.

`delta.J` Scaling parameter for the matrix of smoothing parameters. Only necessary if the test `gofKernel` is part of `tests`.

`nodes.Integration` Number of knots of the bivariate Gauss-Legendre quadrature. Only necessary if the test `gofKernel` is part of `tests`.

`lower` Lower bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default `NULL`.

`upper` Upper bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default `NULL`.

`seed.active` Has to be either an integer or a vector of M+1 integers. If an integer, then the seeds for the bootstrapping procedure will be simulated. If M+1 seeds are provided, then these seeds are used in the bootstrapping procedure. Defaults to `NULL`, then `R` generates the seeds from the computer runtime. Controlling the seeds is useful for reproducibility of a simulation study to compare the power of the tests or for reproducibility of an empirical study.

`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.

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

Note that this function does not display `warning()` messages. Due to the large amount of tests run at once, the messages are not tracable to the situation when they appeared. Hence they are omitted for this function.

Value

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

`method`	a character which informs about the performed analysis
`copula`	the copula tested for
`margins`	the method used to estimate the margin distribution.
`param.margins`	the parameters of the estimated margin distributions. Only applicable if the margins were not specified as `"ranks"` or `NULL`.
`theta`	dependence parameters of the copulae
`df`	the degrees of freedem of the copula. Only applicable for t-copula.
`res.tests`	a matrix with the p-values and test statistics of the hybrid and the individual tests

Examples

```
data(IndexReturns2D)

gof(IndexReturns2D, priority = "tests", copula = "normal",
tests = c("gofRosenblattSnB", "gofRosenblattSnC"), M = 5)
```

`gofArchmChisq` *ArchmChisq Gof test using the Anderson-Darling test statistic and the chi-square distribution*

Description

`gofArchmChisq` contains the Chisq gof test with a Rosenblatt transformation for archimedean copulae, described in Hering and Hofert (2015). The test follows the RosenblattChisq test as 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 (except for the `"amh"` copula) and the possible copulae are `"clayton"`, `"gumbel"`, `"frank"`, `"joe"` and `"amh"`. The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used.

Usage

```
gofArchmChisq(
  copula = c("clayton", "gumbel", "frank", "joe", "amh"),
  x,
  param = 0.5,
  param.est = TRUE,
```

```
  margins = "ranks",
  flip = 0,
  M = 1000,
  lower = NULL,
  upper = NULL,
  seed.active = NULL,
  processes = 1
)
```

Arguments

copula	The copula to test for. Possible are "clayton", "gumbel", "frank", "joe" and "amh".
x	A matrix containing the data with rows being observations and columns being variables.
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.
margins	Specifies which estimation method for the margins shall be used. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively the following distributions can be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp"). Input can be either one method, e.g. "ranks", which will be used for estimation of all data sequences. Also an individual method for each margin can be specified, e.g. c("ranks", "norm", "t") for 3 data sequences. If one does not want to estimate the margins, set it to NULL.
flip	The control parameter to flip the copula by 90, 180, 270 degrees clockwise. Only applicable for bivariate copula. Default is 0 and possible inputs are 0, 90, 180, 270 and NULL.
M	Number of bootstrapping loops.
lower	Lower bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
upper	Upper bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
seed.active	Has to be either an integer or a vector of M+1 integers. If an integer, then the seeds for the bootstrapping procedure will be simulated. If M+1 seeds are provided, then these seeds are used in the bootstrapping procedure. Defaults to NULL, then R generates the seeds from the computer runtime. Controlling the seeds is useful for reproducibility of a simulation study to compare the power of the tests or for reproducibility of an empirical study.
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 Anderson-Darling test statistic (supposedly) computes U[0,1]-distributed (under H_0) random variates via the distribution function of chi-square distribution with d degrees of freedom, see Hofert et al. (2014). 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 transformation for archimedean copula which uses the mapping $\mathcal{R} : (0,1)^d \to (0,1)^d$. Following Hering and Hofert (2015) the mapping provides pseudo observations E_i, given by

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

Let C be an Archimedean copula with d monotone generator ψ and continuous Kendall distribution function K_C. Then,

$$e_j = \left(\frac{\sum_{k=1}^{j} \psi^{-1}(u_k)}{\sum_{k=1}^{j+1} \psi^{-1}(u_k)} \right)^j, j \in \{1, \ldots, d-1\}$$

and

$$e_d = \frac{n}{n+1} K_C(C(u))$$

.

The Anderson-Darling test statistic of the variates

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

is computed (via `ADGofTest::ad.test`), where $x_j = \sum_{i=1}^{d} (\Phi^{-1}(e_{ij}))^2$, Φ^{-1} denotes the quantile function of the standard normal distribution function, χ_d^2 denotes the distribution function of the chi-square distribution with d degrees of freedom, and u_{ij} is the jth component in the ith 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 statistc, respectively.

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

Value

An object of the class gofCOP with the components

method	a character which informs about the performed analysis
copula	the copula tested for
margins	the method used to estimate the margin distribution.
param.margins	the parameters of the estimated margin distributions. Only applicable if the margins were not specified as "ranks" or NULL.
theta	dependence parameters of the copulae
df	the degrees of freedem of the copula. Only applicable for t-copula.
res.tests	a matrix with the p-values and test statistics of the hybrid and the individual tests

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*. doi: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

Christian Hering, Marius Hofert (2015) Goodness-of-fit tests for Archimedean copulas in high dimensions. *In: Glau K., Scherer M., Zagst R. (eds) Innovations in Quantitative Risk Management, Springer Proceedings in Mathematics & Statistics, Volume 99, Springer, Cham, 357-373*. doi:10.1007/9783319091143_21

Examples

```
data(IndexReturns2D)

gofArchmChisq("clayton", IndexReturns2D, M = 10)
```

---

gofArchmGamma *The ArchmGamma Gof test using the Anderson-Darling test statistic and the gamma distribution*

Description

gofArchmGamma contains the Gamma gof test with a Rosenblatt transformation for archimedean copulae, described in Hering and Hofert (2015). The test follows the RosenblattChisq test as 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 (except for the "amh" copula) and the possible copulae are "clayton", "gumbel", "frank", "joe" and "amh". The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used.

Usage

```
gofArchmGamma(
  copula = c("clayton", "gumbel", "frank", "joe", "amh"),
  x,
  param = 0.5,
  param.est = TRUE,
  margins = "ranks",
  flip = 0,
  M = 1000,
  lower = NULL,
  upper = NULL,
  seed.active = NULL,
  processes = 1
)
```

Arguments

copula	The copula to test for. Possible are "clayton", "gumbel", "frank", "joe" and "amh".
x	A matrix containing the data with rows being observations and columns being variables.
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.
margins	Specifies which estimation method for the margins shall be used. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively the following distributions can be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp"). Input can be either one method, e.g. "ranks", which will be used for estimation of all data sequences. Also an individual method for each margin can be specified, e.g. c("ranks", "norm", "t") for 3 data sequences. If one does not want to estimate the margins, set it to NULL.

flip	The control parameter to flip the copula by 90, 180, 270 degrees clockwise. Only applicable for bivariate copula. Default is 0 and possible inputs are 0, 90, 180, 270 and NULL.
M	Number of bootstrapping loops.
lower	Lower bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
upper	Upper bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
seed.active	Has to be either an integer or a vector of M+1 integers. If an integer, then the seeds for the bootstrapping procedure will be simulated. If M+1 seeds are provided, then these seeds are used in the bootstrapping procedure. Defaults to NULL, then R generates the seeds from the computer runtime. Controlling the seeds is useful for reproducibility of a simulation study to compare the power of the tests or for reproducibility of an empirical study.
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 Anderson-Darling test statistic (supposedly) computes U[0,1]-distributed (under H_0) random variates via the distribution function of the gamma distribution, see Hofert et al. (2014). 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 transformation for archimedean copula which uses the mapping $\mathcal{R} : (0,1)^d \to (0,1)^d$. Following Hering and Hofert (2015) the mapping provides pseudo observations E_i, given by

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

Let C be an Archimedean copula with d monotone generator ψ and continuous Kendall distribution function K_C. Then,

$$e_j = \left(\frac{\sum_{k=1}^{j} \psi^{-1}(u_k)}{\sum_{k=1}^{j+1} \psi^{-1}(u_k)} \right)^j, j \in \{1, \ldots, d-1\}$$

and

$$e_d = \frac{n}{n+1} K_C(C(u))$$

.

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 statistc, respectively.

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

Value

An object of the `class` gofCOP with the components

`method`	a character which informs about the performed analysis
`copula`	the copula tested for
`margins`	the method used to estimate the margin distribution.
`param.margins`	the parameters of the estimated margin distributions. Only applicable if the margins were not specified as `"ranks"` or `NULL`.
`theta`	dependence parameters of the copulae
`df`	the degrees of freedem of the copula. Only applicable for t-copula.
`res.tests`	a matrix with the p-values and test statistics of the hybrid and the individual tests

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*. doi: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

Christian Hering, Marius Hofert (2015) Goodness-of-fit tests for Archimedean copulas in high dimensions. *In: Glau K., Scherer M., Zagst R. (eds) Innovations in Quantitative Risk Management, Springer Proceedings in Mathematics & Statistics, Volume 99, Springer, Cham, 357-373*. doi:10.1007/9783319091143_21

Examples

```
data(IndexReturns2D)

gofArchmGamma("clayton", IndexReturns2D, M = 10)
```

`gofArchmSnB` *The ArchmSnB test based on the Rosenblatt transformation for archimedean copula*

Description

`gofArchmSnB` contains the SnB gof test with a Rosenblatt transformation for archimedean copulae, described in Hering and Hofert (2015). The test follows the RosenblattChisq test as 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 (except for the `"amh"` copula) and the possible copulae are `"clayton"`, `"gumbel"`, `"frank"`, `"joe"` and `"amh"`. The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used.

Usage

```
gofArchmSnB(
  copula = c("clayton", "gumbel", "frank", "joe", "amh"),
  x,
  param = 0.5,
  param.est = TRUE,
  margins = "ranks",
  flip = 0,
  M = 1000,
  lower = NULL,
  upper = NULL,
  seed.active = NULL,
  processes = 1
)
```

Arguments

`copula`	The copula to test for. Possible are `"clayton"`, `"gumbel"`, `"frank"`, `"joe"` and `"amh"`.
`x`	A matrix containing the data with rows being observations and columns being variables.

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.
margins	Specifies which estimation method for the margins shall be used. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively the following distributions can be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp"). Input can be either one method, e.g. "ranks", which will be used for estimation of all data sequences. Also an individual method for each margin can be specified, e.g. c("ranks", "norm", "t") for 3 data sequences. If one does not want to estimate the margins, set it to NULL.
flip	The control parameter to flip the copula by 90, 180, 270 degrees clockwise. Only applicable for bivariate copula. Default is 0 and possible inputs are 0, 90, 180, 270 and NULL.
M	Number of bootstrapping loops.
lower	Lower bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
upper	Upper bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
seed.active	Has to be either an integer or a vector of M+1 integers. If an integer, then the seeds for the bootstrapping procedure will be simulated. If M+1 seeds are provided, then these seeds are used in the bootstrapping procedure. Defaults to NULL, then R generates the seeds from the computer runtime. Controlling the seeds is useful for reproducibility of a simulation study to compare the power of the tests or for reproducibility of an empirical study.
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 transformation for archimedean copula which uses the mapping $\mathcal{R} : (0,1)^d \to (0,1)^d$. Following Hering and Hofert (2015) the mapping provides pseudo observations E_i, given by

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

Let C be an Archimedean copula with d monotone generator ψ and continuous Kendall distribution function K_C. Then,

$$e_j = \left(\frac{\sum_{k=1}^{j} \psi^{-1}(u_k)}{\sum_{k=1}^{j+1} \psi^{-1}(u_k)} \right)^j, j \in \{1, \ldots, d-1\}$$

and

$$e_d = \frac{n}{n+1} K_C(C(u))$$

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

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 statistc, respectively.

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

Value

An object of the `class` gofCOP with the components

`method`	a character which informs about the performed analysis
`copula`	the copula tested for
`margins`	the method used to estimate the margin distribution.
`param.margins`	the parameters of the estimated margin distributions. Only applicable if the margins were not specified as `"ranks"` or `NULL`.
`theta`	dependence parameters of the copulae
`df`	the degrees of freedem of the copula. Only applicable for t-copula.
`res.tests`	a matrix with the p-values and test statistics of the hybrid and the individual tests

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*. doi: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

Christian Hering, Marius Hofert (2015) Goodness-of-fit tests for Archimedean copulas in high dimensions. *In: Glau K., Scherer M., Zagst R. (eds) Innovations in Quantitative Risk Management, Springer Proceedings in Mathematics & Statistics, Volume 99, Springer, Cham, 357-373*. doi:10.1007/9783319091143_21

Examples

```
data(IndexReturns2D)

gofArchmSnB("clayton", IndexReturns2D, M = 10)
```

`gofArchmSnC` *The ArchmSnC test based on the Rosenblatt transformation for archimedean copula*

Description

`gofArchmSnC` contains the SnC gof test with a Rosenblatt transformation for archimedean copulae, described in Hering and Hofert (2015). The test follows the RosenblattChisq test as 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 (except for the `"amh"` copula) and the possible copulae are `"clayton"`, `"gumbel"`, `"frank"`, `"joe"` and `"amh"`. The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used.

Usage

```
gofArchmSnC(
  copula = c("clayton", "gumbel", "frank", "joe", "amh"),
  x,
  param = 0.5,
  param.est = TRUE,
  margins = "ranks",
  flip = 0,
  M = 1000,
  lower = NULL,
  upper = NULL,
  seed.active = NULL,
  processes = 1
)
```

Arguments

`copula`	The copula to test for. Possible are `"clayton"`, `"gumbel"`, `"frank"`, `"joe"` and `"amh"`.
`x`	A matrix containing the data with rows being observations and columns being variables.

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.
margins	Specifies which estimation method for the margins shall be used. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively the following distributions can be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp"). Input can be either one method, e.g. "ranks", which will be used for estimation of all data sequences. Also an individual method for each margin can be specified, e.g. c("ranks", "norm", "t") for 3 data sequences. If one does not want to estimate the margins, set it to NULL.
flip	The control parameter to flip the copula by 90, 180, 270 degrees clockwise. Only applicable for bivariate copula. Default is 0 and possible inputs are 0, 90, 180, 270 and NULL.
M	Number of bootstrapping loops.
lower	Lower bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
upper	Upper bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
seed.active	Has to be either an integer or a vector of M+1 integers. If an integer, then the seeds for the bootstrapping procedure will be simulated. If M+1 seeds are provided, then these seeds are used in the bootstrapping procedure. Defaults to NULL, then R generates the seeds from the computer runtime. Controlling the seeds is useful for reproducibility of a simulation study to compare the power of the tests or for reproducibility of an empirical study.
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 transformation for archimedean copula which uses the mapping $\mathcal{R} : (0,1)^d \to (0,1)^d$. Following Hering and Hofert (2015) the mapping provides pseudo observations E_i, given by

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

Let C be an Archimedean copula with d monotone generator ψ and continuous Kendall distribution function K_C. Then,

$$e_j = \left(\frac{\sum_{k=1}^{j} \psi^{-1}(u_k)}{\sum_{k=1}^{j+1} \psi^{-1}(u_k)} \right)^j, j \in \{1, \ldots, d-1\}$$

and

$$e_d = \frac{n}{n+1} K_C(C(u))$$

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

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 statistc, respectively.

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

Value

An object of the `class` gofCOP with the components

`method`	a character which informs about the performed analysis
`copula`	the copula tested for
`margins`	the method used to estimate the margin distribution.
`param.margins`	the parameters of the estimated margin distributions. Only applicable if the margins were not specified as `"ranks"` or `NULL`.
`theta`	dependence parameters of the copulae
`df`	the degrees of freedem of the copula. Only applicable for t-copula.
`res.tests`	a matrix with the p-values and test statistics of the hybrid and the individual tests

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*. doi: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

Christian Hering, Marius Hofert (2015) Goodness-of-fit tests for Archimedean copulas in high dimensions. *In: Glau K., Scherer M., Zagst R. (eds) Innovations in Quantitative Risk Management, Springer Proceedings in Mathematics & Statistics, Volume 99, Springer, Cham, 357-373*. doi:10.1007/9783319091143_21

Examples

```
data(IndexReturns2D)

gofArchmSnC("clayton", IndexReturns2D, M = 10)
```

---

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 takes.

Usage

```
gofCheckTime(
  copula,
  x,
  tests = NULL,
  customTests = NULL,
  param = 0.5,
  param.est = TRUE,
  df = 4,
  df.est = TRUE,
  margins = "ranks",
  flip = 0,
  M = 1000,
  MJ = 100,
  dispstr = "ex",
  print.res = TRUE,
  m = 1,
  delta.J = 0.5,
  nodes.Integration = 12,
  lower = NULL,
  upper = NULL,
  seed.active = NULL,
  processes = 1
)
```

Arguments

copula	A character vector which indicates the copula to test for. Possible are "normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos", "huslerReiss", "tawn", "tev", "fgm" and "plackett".
x	A matrix containing the data with rows being observations and columns being variables.

tests	A character vector which indicates the test to use.
customTests	A character vector which indicates the customized test to use, if any.
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. For the "gofPIOSTn" test the entry is limited to 60 degrees of freedom for computational reasons.
df.est	Indicates if df shall be estimated. Has to be either FALSE or TRUE, where TRUE means that it will be estimated. For the "gofPIOSTn" test the estimate is limited to 60 degrees of freedom for computational reasons.
margins	Specifies which estimation method for the margins shall be used. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively the following distributions can be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp"). Input can be either one method, e.g. "ranks", which will be used for estimation of all data sequences. Also an individual method for each margin can be specified, e.g. c("ranks", "norm", "t") for 3 data sequences. If one does not want to estimate the margins, set it to NULL.
flip	The control parameter to flip the copula by 90, 180, 270 degrees clockwise. Only applicable for bivariate copula. Default is 0 and possible inputs are 0, 90, 180, 270 and NULL.
M	The number of bootstrapping rounds which shall be later taken in the estimation.
MJ	Just for the test gofKernel. Size of bootstrapping sample.
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.
print.res	Logical which defines if the resulting time shall be printed or given as value. Default is TRUE.
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.
lower	Lower bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
upper	Upper bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
seed.active	Has to be either an integer or a vector of M+1 integers. If an integer, then the seeds for the bootstrapping procedure will be simulated. If M+1 seeds are

provided, then these seeds are used in the bootstrapping procedure. Defaults to NULL, then R generates the seeds from the computer runtime. Controlling the seeds is useful for reproducibility of a simulation study to compare the power of the tests or for reproducibility of an empirical study.

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

```
## Not run:
data(IndexReturns2D)

gofCheckTime("normal", IndexReturns2D, "gofKendallKS", M = 10000)

## End(Not run)
```

---

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,
  tests = c("gofPIOSRn", "gofKernel"),
  customTests = NULL,
  margins = "ranks",
  flip = 0,
  M = 1000,
  MJ = 100,
  dispstr = "ex",
  m = 1,
  delta.J = 0.5,
  nodes.Integration = 12,
  lower = NULL,
  upper = NULL,
  seed.active = NULL,
  processes = 1
)
```

Arguments

copulaobject	An object with of class copula from the copula package.
x	A matrix containing the data with rows being observations and columns being variables.
tests	A character vector which indicates the tests to use. Possible choices are the individual tests implemented in this package.
customTests	A character vector which indicates the customized test to use, if any. The test has to be loaded into the workspace. Currently the function containing the test has to have 2 arguments, the first one for the dataset and the second one for the copula to test for. The arguments have to be named "x" and "copula" respectively.
margins	Specifies which estimation method for the margins shall be used. 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"). Input can be either one method, e.g. "ranks", which will be used for estimation of all data sequences. Also an individual method for each margin can be specified, e.g. c("ranks", "norm", "t") for 3 data sequences. If one does not want to estimate the margins, set it to NULL.
flip	The control parameter to flip the copula by 90, 180, 270 degrees clockwise. Only applicable for bivariate copula. Default is 0 and possible inputs are 0, 90, 180, 270 and NULL.
M	Number of bootstrapping samples in the single tests.
MJ	Size of bootstrapping sample. Only necessary if the test gofKernel is part of testset.
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. 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.
lower	Lower bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
upper	Upper bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
seed.active	Has to be either an integer or a vector of M+1 integers. If an integer, then the seeds for the bootstrapping procedure will be simulated. If M+1 seeds are provided, then these seeds are used in the bootstrapping procedure. Defaults to NULL, then R generates the seeds from the computer runtime. Controlling the

seeds is useful for reproducibility of a simulation study to compare the power of the tests or for reproducibility of an empirical study.

`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 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.

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

Value

An object of the `class` gofCOP with the components

`method`	a character which informs about the performed analysis
`copula`	the copula tested for
`margins`	the method used to estimate the margin distribution.
`param.margins`	the parameters of the estimated margin distributions. Only applicable if the margins were not specified as `"ranks"` or `NULL`.
`theta`	dependence parameters of the copulae
`df`	the degrees of freedem of the copula. Only applicable for t-copula.
`res.tests`	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(IndexReturns2D)
copObject = normalCopula(param = 0.5)

gofco(copObject, x = IndexReturns2D, tests = c("gofPIOSRn", "gofKernel"),
      M = 20)
```

gofCopula4Test *Implemented copula for a certain test*

Description

gofCopula4Test returns for a given test the applicable implemented copula.

Usage

```
gofCopula4Test(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

```
gofCopula4Test("gofRosenblattSnB")

gofCopula4Test("gofPIOSTn")
```

gofCustomTest *Function to derive own tests*

Description

gofCustomTest allows to include own Goodness-of-Fit tests and perform the test with the package. 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. It is possible to insert datasets of all dimensions above 1 and the possible copulae are "normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos", "huslerReiss", "tawn", "tev", "fgm" and "plackett". The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used.

Usage

```
gofCustomTest(
 copula = c("normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos",
    "huslerReiss", "tawn", "tev", "fgm", "plackett"),
  x,
  customTest = NULL,
  param = 0.5,
  param.est = TRUE,
  df = 4,
  df.est = TRUE,
  margins = "ranks",
  flip = 0,
  M = 1000,
  dispstr = "ex",
  lower = NULL,
  upper = NULL,
  seed.active = NULL,
  processes = 1
)
```

Arguments

copula	The copula to test for. Possible are "normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos", "huslerReiss", "tawn", "tev", "fgm" and "plackett".
x	A matrix containing the data with rows being observations and columns being variables.
customTest	A character string with the name of the customized test. The test has to be loaded into the workspace. Currently the function containing the test has to have 2 arguments, the first one for the dataset and the second one for the copula to test for. The arguments have to be named "x" and "copula" respectively.
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 for the margins shall be used. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively the following distributions can be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp"). Input can be either one method, e.g. "ranks", which will be used for estimation of all data sequences. Also an individual method for each margin can be specified, e.g. c("ranks", "norm", "t") for 3 data sequences. If one does not want to estimate the margins, set it to NULL.

flip	The control parameter to flip the copula by 90, 180, 270 degrees clockwise. Only applicable for bivariate copula. Default is 0 and possible inputs are 0, 90, 180, 270 and NULL.
M	Number of bootstrapping loops.
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.
lower	Lower bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
upper	Upper bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
seed.active	Has to be either an integer or a vector of M+1 integers. If an integer, then the seeds for the bootstrapping procedure will be simulated. If M+1 seeds are provided, then these seeds are used in the bootstrapping procedure. Defaults to NULL, then R generates the seeds from the computer runtime. Controlling the seeds is useful for reproducibility of a simulation study to compare the power of the tests or for reproducibility of an empirical study.
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 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 statistc, respectively.

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

Value

An object of the class gofCOP with the components

method	a character which informs about the performed analysis
copula	the copula tested for
margins	the method used to estimate the margin distribution.
param.margins	the parameters of the estimated margin distributions. Only applicable if the margins were not specified as "ranks" or NULL.
theta	dependence parameters of the copulae
df	the degrees of freedem of the copula. Only applicable for t-copula.
res.tests	a matrix with the p-values and test statistics of the hybrid and the individual tests

Examples

```
# For illustration we load here the test statistic of the gofSn test
Testfunc = function(x, copula) {
            C.theo = pCopula(x, copula = copula)
            C.n = F.n(x, X = x)
            CnK = sum((C.n - C.theo)^2)
            return(CnK)
}

data(IndexReturns2D)

gofCustomTest(copula = "normal", x = IndexReturns2D,
              customTest = "Testfunc", M=10)
```

---

gofCvM *The CvM gof test using the empirical copula*

Description

gofCvM performs the "CvM" 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", "clayton", "gumbel", "frank", "joe", "amh", "galambos", "huslerReiss", "tawn", "tev", "fgm" and "plackett". The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used.

Usage

```
gofCvM(
 copula = c("normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos",
    "huslerReiss", "tawn", "tev", "fgm", "plackett"),
  x,
  param = 0.5,
  param.est = TRUE,
  df = 4,
  df.est = TRUE,
  margins = "ranks",
  flip = 0,
  M = 1000,
  dispstr = "ex",
  lower = NULL,
  upper = NULL,
```

```
  seed.active = NULL,
  processes = 1
)
```

Arguments

copula	The copula to test for. Possible are "normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos", "huslerReiss", "tawn", "tev", "fgm" and "plackett".
x	A matrix containing the data with rows being observations and columns being variables.
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 for the margins shall be used. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively the following distributions can be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp"). Input can be either one method, e.g. "ranks", which will be used for estimation of all data sequences. Also an individual method for each margin can be specified, e.g. c("ranks", "norm", "t") for 3 data sequences. If one does not want to estimate the margins, set it to NULL.
flip	The control parameter to flip the copula by 90, 180, 270 degrees clockwise. Only applicable for bivariate copula. Default is 0 and possible inputs are 0, 90, 180, 270 and NULL.
M	Number of bootstrapping loops.
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.
lower	Lower bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
upper	Upper bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
seed.active	Has to be either an integer or a vector of M+1 integers. If an integer, then the seeds for the bootstrapping procedure will be simulated. If M+1 seeds are provided, then these seeds are used in the bootstrapping procedure. Defaults to NULL, then R generates the seeds from the computer runtime. Controlling the seeds is useful for reproducibility of a simulation study to compare the power of the tests or for reproducibility of an empirical study.

`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 $i = 1, \ldots, n$, $j = 1, \ldots, d$ and $\mathbf{u} \in [0,1]^d$ is the empirical copula given by $C_n(\mathbf{u}) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{I}(U_{i1} \leq u_1, \ldots, U_{id} \leq u_d)$. 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 statistc, respectively.

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

Value

An object of the `class` gofCOP with the components

`method`	a character which informs about the performed analysis
`copula`	the copula tested for
`margins`	the method used to estimate the margin distribution.
`param.margins`	the parameters of the estimated margin distributions. Only applicable if the margins were not specified as `"ranks"` or `NULL`.
`theta`	dependence parameters of the copulae
`df`	the degrees of freedem of the copula. Only applicable for t-copula.
`res.tests`	a matrix with the p-values and test statistics of the hybrid and the individual tests

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(IndexReturns2D)

gofCvM("normal", IndexReturns2D, M = 10)
```

`gofGetHybrid` *GetHybrid gof test*

Description

`gofGetHybrid` computes based on previous test results from this package and on p-values from your own goodness-of-fit tests the hybrid test p-values for the specified testing size.

Usage

```
gofGetHybrid(result, p_values = NULL, nsets = NULL)
```

Arguments

`result`	An object of `class` gofCOP.
`p_values`	A vector containing different p-values from your own goodness-of-fit tests. If the elements are unnamed, the test results in the output will be labeled with Test_A, Test_B etc.
`nsets`	The desired number of tests to be included in each hybrid test. It should be an integer larger than 1 and smaller or equal than the number of tests given in `result` and `p_values` together. If `nsets` is set NULL (default), all possible testing sizes are calculated.

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 dominant 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:

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

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.

Value

An object of the class gofCOP with the components

method	a character which informs about the performed analysis
copula	the copula tested for
margins	the method used to estimate the margin distribution.
param.margins	the parameters of the estimated margin distributions. Only applicable if the margins were not specified as "ranks" or NULL.
theta	dependence parameters of the copulae
df	the degrees of freedem of the copula. Only applicable for t-copula.
res.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* doi:10.1016/j.jeconom.2016.02.017

Examples

```
data(IndexReturns2D)

res_2 = gof(x = IndexReturns2D, copula = "normal",
            tests = c("gofKernel", "gofKendallCvM", "gofWhite"), M = 10)
gofGetHybrid(result = res_2,
             p_values = c("MyTest" = 0.3, "AnotherTest" = 0.7), nsets = 3)
```

---

gofKendallCvM *gof test (Cramer-von Mises) based on Kendall's process*

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", "clayton", "gumbel", "frank", "joe", "amh", "galambos", "huslerReiss", "tawn", "tev", "fgm" and "plackett". 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 = c("normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos",
    "huslerReiss", "tawn", "tev", "fgm", "plackett"),
  x,
  param = 0.5,
  param.est = TRUE,
  df = 4,
  df.est = TRUE,
  margins = "ranks",
  flip = 0,
  M = 1000,
  dispstr = "ex",
  lower = NULL,
  upper = NULL,
  seed.active = NULL,
  processes = 1
)
```

Arguments

copula	The copula to test for. Possible are "normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos", "huslerReiss", "tawn", "tev", "fgm" and "plackett".
x	A matrix containing the data with rows being observations and columns being variables.
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 for the margins shall be used. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively the following distributions can be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp"). Input can be either one method, e.g. "ranks", which will be used for estimation of all data sequences. Also an individual method for each margin can be specified, e.g. c("ranks", "norm", "t") for 3 data sequences. If one does not want to estimate the margins, set it to NULL.
flip	The control parameter to flip the copula by 90, 180, 270 degrees clockwise. Only applicable for bivariate copula. Default is 0 and possible inputs are 0, 90, 180, 270 and NULL.
M	Number of bootstrap samples.

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.
lower	Lower bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
upper	Upper bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
seed.active	Has to be either an integer or a vector of M+1 integers. If an integer, then the seeds for the bootstrapping procedure will be simulated. If M+1 seeds are provided, then these seeds are used in the bootstrapping procedure. Defaults to NULL, then R generates the seeds from the computer runtime. Controlling the seeds is useful for reproducibility of a simulation study to compare the power of the tests or for reproducibility of an empirical study.
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 $i = 1, \ldots, n$, $j = 1, \ldots, d$ and $\mathbf{u} \in [0,1]^d$ is the empirical copula given by $C_n(\mathbf{u}) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{I}(U_{i1} \leq u_1, \ldots, U_{id} \leq u_d)$. Let the rescaled pseudo observations be $V_1 = C_n(U_1), \ldots, V_n = C_n(U_n)$ 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 Θ 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 H_0 the test is 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 parallelisation via processes does not make sense. The registration of the parallel processes increases the computation time. Please consider to enable parallelisation just for high values of M.

Value

An object of the class gofCOP with the components

method	a character which informs about the performed analysis
copula	the copula tested for
margins	the method used to estimate the margin distribution.
param.margins	the parameters of the estimated margin distributions. Only applicable if the margins were not specified as "ranks" or NULL.
theta	dependence parameters of the copulae
df	the degrees of freedem of the copula. Only applicable for t-copula.
res.tests	a matrix with the p-values and test statistics of the hybrid and the individual tests

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*. doi:10.1016/j.insmatheco.2007.10.005

Christian Genest, Jean-Francois Quessy, Bruno Remillard (2006). Goodness-of-fit Procedures for Copula Models Based on the Probability Integral Transformation. *Scandinavian Journal of Statistics, Volume 33, Issue 2, 2006, Pages 337-366*. doi:10.1111/j.14679469.2006.00470.x

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(IndexReturns2D)

gofKendallCvM("normal", IndexReturns2D, M = 10)
```

---

gofKendallKS *gof test (Kolmogorov-Smirnov) based on Kendall's process*

Description

gofKendallKS tests a given dataset for a copula based on Kendall's process with the Kolmogorov-Smirnov 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", "clayton", "gumbel", "frank", "joe", "amh", "galambos", "huslerReiss", "tawn", "tev", "fgm" and "plackett". 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 = c("normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos",
    "huslerReiss", "tawn", "tev", "fgm", "plackett"),
  x,
  param = 0.5,
  param.est = TRUE,
  df = 4,
  df.est = TRUE,
  margins = "ranks",
  flip = 0,
  M = 1000,
  dispstr = "ex",
  lower = NULL,
  upper = NULL,
  seed.active = NULL,
  processes = 1
)
```

Arguments

copula	Possible are "normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos", "huslerReiss", "tawn", "tev", "fgm" and "plackett".
x	A matrix containing the data with rows being observations and columns being variables.
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 for the margins shall be used. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively the following distributions can be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp"). Input can be either one method, e.g. "ranks", which will be used for estimation of all data sequences. Also an individual method for each margin can be specified, e.g. c("ranks", "norm", "t") for 3 data sequences. If one does not want to estimate the margins, set it to NULL.
flip	The control parameter to flip the copula by 90, 180, 270 degrees clockwise. Only applicable for bivariate copula. Default is 0 and possible inputs are 0, 90, 180, 270 and NULL.
M	Number of bootstrap samples.

`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`.

`lower` Lower bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default `NULL`.

`upper` Upper bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default `NULL`.

`seed.active` Has to be either an integer or a vector of M+1 integers. If an integer, then the seeds for the bootstrapping procedure will be simulated. If M+1 seeds are provided, then these seeds are used in the bootstrapping procedure. Defaults to `NULL`, then `R` generates the seeds from the computer runtime. Controlling the seeds is useful for reproducibility of a simulation study to compare the power of the tests or for reproducibility of an empirical study.

`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 $i = 1, \ldots, n$, $j = 1, \ldots, d$ and $\mathbf{u} \in [0,1]^d$ is the empirical copula given by $C_n(\mathbf{u}) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{I}(U_{i1} \leq u_1, \ldots, U_{id} \leq u_d)$. Let the rescaled pseudo observations be $V_1 = C_n(U_1), \ldots, V_n = C_n(U_n)$ 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 Θ being an open subset of R^p for an integer $p \geq 1$, see Genest et al. (2009). The resulting Kolmogorov-Smirnof 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 parallelisation via `processes` does not make sense. The registration of the parallel processes increases the computation time. Please consider to enable parallelisation just for high values of `M`.

Value

An object of the class gofCOP with the components

method	a character which informs about the performed analysis
copula	the copula tested for
margins	the method used to estimate the margin distribution.
param.margins	the parameters of the estimated margin distributions. Only applicable if the margins were not specified as "ranks" or NULL.
theta	dependence parameters of the copulae
df	the degrees of freedem of the copula. Only applicable for t-copula.
res.tests	a matrix with the p-values and test statistics of the hybrid and the individual tests

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*. doi:10.1016/j.insmatheco.2007.10.005

Christian Genest, Jean-Francois Quessy, Bruno Remillard (2006). Goodness-of-fit Procedures for Copula Models Based on the Probability Integral Transformation. *Scandinavian Journal of Statistics, Volume 33, Issue 2, 2006, Pages 337-366*. doi:10.1111/j.14679469.2006.00470.x

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(IndexReturns2D)

gofKendallKS("normal", IndexReturns2D, M = 10)
```

---

gofKernel *2 dimensional gof test of Scaillet (2007)*

Description

gofKernel tests a 2 dimensional dataset with the Scaillet test for a copula. The possible copulae are "normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos", "huslerReiss", "tawn", "tev", "fgm" and "plackett". 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 = c("normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos",
    "huslerReiss", "tawn", "tev", "fgm", "plackett"),
  x,
  param = 0.5,
  param.est = TRUE,
  df = 4,
  df.est = TRUE,
  margins = "ranks",
  flip = 0,
  M = 1000,
  MJ = 100,
  dispstr = "ex",
  delta.J = 0.5,
  nodes.Integration = 12,
  lower = NULL,
  upper = NULL,
  seed.active = NULL,
  processes = 1
)
```

Arguments

copula	The copula to test for. Possible are "normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos", "huslerReiss", "tawn", "tev", "fgm" and "plackett".
x	A matrix containing the data with rows being observations and columns being variables.
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 for the margins shall be used. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively the following distributions can be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp"). Input can be either one method, e.g. "ranks", which will be used for estimation of all data sequences. Also an individual method for each margin can be specified, e.g. c("ranks", "norm", "t") for 3 data sequences. If one does not want to estimate the margins, set it to NULL.

`flip` The control parameter to flip the copula by 90, 180, 270 degrees clockwise. Only applicable for bivariate copula. Default is 0 and possible inputs are 0, 90, 180, 270 and NULL.

`M` Number of bootstrapping loops.

`MJ` Size of bootstrapping sample.

`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`.

`delta.J` Scaling parameter for the matrix of smoothing parameters.

`nodes.Integration`
Number of knots of the bivariate Gauss-Legendre quadrature.

`lower` Lower bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default `NULL`.

`upper` Upper bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default `NULL`.

`seed.active` Has to be either an integer or a vector of M+1 integers. If an integer, then the seeds for the bootstrapping procedure will be simulated. If M+1 seeds are provided, then these seeds are used in the bootstrapping procedure. Defaults to `NULL`, then `R` generates the seeds from the computer runtime. Controlling the seeds is useful for reproducibility of a simulation study to compare the power of the tests or for reproducibility of an empirical study.

`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

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

where U_{ij} for $i = 1, \ldots, n$; $j = 1, \ldots, d$ 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 $H = 2.6073 n^{-1/6} \hat{\Sigma}^{1/2}$ 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 ω 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 parallelisation via `processes` does not make sense. The registration of the parallel processes increases the computation time. Please consider to enable parallelisation just for high values of `M`.

Value

An object of the `class` gofCOP with the components

`method`	a character which informs about the performed analysis
`copula`	the copula tested for
`margins`	the method used to estimate the margin distribution.
`param.margins`	the parameters of the estimated margin distributions. Only applicable if the margins were not specified as `"ranks"` or `NULL`.
`theta`	dependence parameters of the copulae
`df`	the degrees of freedem of the copula. Only applicable for t-copula.
`res.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* doi:10.1016/j.jeconom.2016.02.017

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

Examples

```
data(IndexReturns2D)

gofKernel("normal", IndexReturns2D, M = 5, MJ = 5)
```

gofKS *The KS gof test using the empirical copula*

Description

gofKS performs the "KS" gof test 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", "clayton", "gumbel", "frank", "joe", "amh", "galambos", "huslerReiss", "tawn", "tev", "fgm" and "plackett". The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used.

Usage

```
gofKS(
 copula = c("normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos",
    "huslerReiss", "tawn", "tev", "fgm", "plackett"),
 x,
 param = 0.5,
 param.est = TRUE,
 df = 4,
 df.est = TRUE,
 margins = "ranks",
 flip = 0,
 M = 1000,
 dispstr = "ex",
 lower = NULL,
 upper = NULL,
 seed.active = NULL,
 processes = 1
)
```

Arguments

copula	The copula to test for. Possible are "normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos", "huslerReiss", "tawn", "tev", "fgm" and "plackett".
x	A matrix containing the data with rows being observations and columns being variables.
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 for the margins shall be used. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively the following distributions can be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp"). Input can be either one method, e.g. "ranks", which will be used for estimation of all data sequences. Also an individual method for each margin can be specified, e.g. c("ranks", "norm", "t") for 3 data sequences. If one does not want to estimate the margins, set it to NULL.
flip	The control parameter to flip the copula by 90, 180, 270 degrees clockwise. Only applicable for bivariate copula. Default is 0 and possible inputs are 0, 90, 180, 270 and NULL.
M	Number of bootstrapping loops.
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.
lower	Lower bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
upper	Upper bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
seed.active	Has to be either an integer or a vector of M+1 integers. If an integer, then the seeds for the bootstrapping procedure will be simulated. If M+1 seeds are provided, then these seeds are used in the bootstrapping procedure. Defaults to NULL, then R generates the seeds from the computer runtime. Controlling the seeds is useful for reproducibility of a simulation study to compare the power of the tests or for reproducibility of an empirical study.
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 $i = 1, \ldots, n$, $j = 1, \ldots, d$ and $\mathbf{u} \in [0,1]^d$ is the empirical copula given by $C_n(\mathbf{u}) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{I}(U_{i1} \leq u_1, \ldots, U_{id} \leq u_d)$. 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 resulting Kolmogorov-Smirnof test statistic is then given by

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

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 statistc, respectively.

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

Value

An object of the `class` gofCOP with the components

`method`	a character which informs about the performed analysis
`copula`	the copula tested for
`margins`	the method used to estimate the margin distribution.
`param.margins`	the parameters of the estimated margin distributions. Only applicable if the margins were not specified as `"ranks"` or `NULL`.
`theta`	dependence parameters of the copulae
`df`	the degrees of freedem of the copula. Only applicable for t-copula.
`res.tests`	a matrix with the p-values and test statistics of the hybrid and the individual tests

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(IndexReturns2D)

gofKS("normal", IndexReturns2D, M = 10)
```

`gofOutputHybrid` *Output Hybrid gof test*

Description

`gofOutputHybrid` outputs the desired Hybrid tests from previous test results from this package for the specified testing size.

Usage

```
gofOutputHybrid(result, tests = NULL, nsets = NULL)
```

Arguments

`result`	An object of `class` gofCOP.
`tests`	Individual tests which should be used in the hybrid test. Submit a vector containing the position of the individual tests as they appear in the object submitted, e.g. `c(1,4)` for the 1st and 4th tests. If `tests` is set NULL (default), all possible testing sizes are returned.
`nsets`	The desired number of tests to be included in each hybrid test. It should be an integer larger than 1 and smaller or equal than the number of tests given in `result`. If `nsets` is set NULL (default), all possible testing sizes are calculated.

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 dominant 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:

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

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.

Value

An object of the `class` gofCOP with the components

`method`	a character which informs about the performed analysis
`copula`	the copula tested for
`margins`	the method used to estimate the margin distribution.
`param.margins`	the parameters of the estimated margin distributions. Only applicable if the margins were not specified as `"ranks"` or `NULL`.

theta	dependence parameters of the copulae
df	the degrees of freedem of the copula. Only applicable for t-copula.
res.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* doi:10.1016/j.jeconom.2016.02.017

Examples

```
data(IndexReturns2D)

res1 = gof(IndexReturns2D, priority = "tests", copula = "normal",
           tests = c("gofKendallCvM", "gofRosenblattSnC", "gofKendallKS"),
           M = 5)
gofOutputHybrid(res1, tests = 1, nsets = 2)
# mind the difference to the regular output
res1
```

---

gofPIOSRn *2 and 3 dimensional gof test based on the in-and-out-of-sample approach*

Description

gofPIOSRn tests a 2 or 3 dimensional dataset with the approximate PIOS test for a copula. The possible copulae are "normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos", "fgm" and "plackett". 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 = c("normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos",
    "fgm", "plackett"),
 x,
 param = 0.5,
 param.est = TRUE,
 df = 4,
 df.est = TRUE,
```

```
  margins = "ranks",
  flip = 0,
  M = 1000,
  dispstr = "ex",
  lower = NULL,
  upper = NULL,
  seed.active = NULL,
  processes = 1
)
```

Arguments

copula	The copula to test for. Possible are "normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos", "fgm" and "plackett".
x	A matrix containing the data with rows being observations and columns being variables.
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 for the margins shall be used. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively the following distributions can be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp"). Input can be either one method, e.g. "ranks", which will be used for estimation of all data sequences. Also an individual method for each margin can be specified, e.g. c("ranks", "norm", "t") for 3 data sequences. If one does not want to estimate the margins, set it to NULL.
flip	The control parameter to flip the copula by 90, 180, 270 degrees clockwise. Only applicable for bivariate copula. Default is 0 and possible inputs are 0, 90, 180, 270 and NULL.
M	Number of bootstrapping loops.
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.
lower	Lower bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
upper	Upper bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.

`seed.active` Has to be either an integer or a vector of M+1 integers. If an integer, then the seeds for the bootstrapping procedure will be simulated. If M+1 seeds are provided, then these seeds are used in the bootstrapping procedure. Defaults to `NULL`, then `R` generates the seeds from the computer runtime. Controlling the seeds is useful for reproducibility of a simulation study to compare the power of the tests or for reproducibility of an empirical study.

`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 $i = 1, \ldots, n$; $j = 1, \ldots, d$ and

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

and

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

By defining two information matrices

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

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

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 $n \to \infty$.

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

Value

An object of the class gofCOP with the components

method	a character which informs about the performed analysis
copula	the copula tested for
margins	the method used to estimate the margin distribution.
param.margins	the parameters of the estimated margin distributions. Only applicable if the margins were not specified as "ranks" or NULL.
theta	dependence parameters of the copulae
df	the degrees of freedem of the copula. Only applicable for t-copula.
res.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* doi:10.1016/j.jeconom.2016.02.017

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(IndexReturns2D)

gofPIOSRn("normal", IndexReturns2D, M = 10)
```

---

gofPIOSTn — *2 and 3 dimensional gof test based on the in-and-out-of-sample approach*

Description

gofPIOSTn tests a 2 or 3 dimensional dataset with the PIOS test for a copula. The possible copulae are "normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos", "fgm" and "plackett". 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 = c("normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos",
    "fgm", "plackett"),
  x,
  param = 0.5,
  param.est = TRUE,
  df = 4,
  df.est = TRUE,
  margins = "ranks",
  flip = 0,
  M = 1000,
  dispstr = "ex",
  m = 1,
  lower = NULL,
  upper = NULL,
  seed.active = NULL,
  processes = 1
)
```

Arguments

copula	The copula to test for. Possible are "normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos", "fgm" and "plackett".
x	A matrix containing the data with rows being observations and columns being variables.
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. For computational reasons the entry is limited to 60 degrees of freedom.
df.est	Indicates if df shall be estimated. Has to be either FALSE or TRUE, where TRUE means that it will be estimated. For computational reasons the estimate is limited to 60 degrees of freedom.
margins	Specifies which estimation method for the margins shall be used. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively the following distributions can be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp"). Input can be either one method, e.g. "ranks", which will be used for estimation of all data sequences. Also an individual method for each margin can be specified, e.g. c("ranks", "norm", "t") for 3 data sequences. If one does not want to estimate the margins, set it to NULL.
flip	The control parameter to flip the copula by 90, 180, 270 degrees clockwise. Only applicable for bivariate copula. Default is 0 and possible inputs are 0, 90, 180, 270 and NULL.

M	Number of bootstrapping loops.
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.
lower	Lower bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
upper	Upper bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
seed.active	Has to be either an integer or a vector of M+1 integers. If an integer, then the seeds for the bootstrapping procedure will be simulated. If M+1 seeds are provided, then these seeds are used in the bootstrapping procedure. Defaults to NULL, then R generates the seeds from the computer runtime. Controlling the seeds is useful for reproducibility of a simulation study to compare the power of the tests or for reproducibility of an empirical study.
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 blocks of length m are constructed 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 $i = 1, \ldots, n$; $j = 1, \ldots, d$ and

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

and

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

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

Value

An object of the class gofCOP with the components

method	a character which informs about the performed analysis
copula	the copula tested for
margins	the method used to estimate the margin distribution.
param.margins	the parameters of the estimated margin distributions. Only applicable if the margins were not specified as "ranks" or NULL.
theta	dependence parameters of the copulae
df	the degrees of freedem of the copula. Only applicable for t-copula.
res.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* doi:10.1016/j.jeconom.2016.02.017

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(IndexReturns2D)

gofPIOSTn("normal", IndexReturns2D, M = 10)
```

---

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", "clayton", "gumbel", "frank", "joe", "amh", "galambos", "fgm" and "plackett". 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 = c("normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos",
    "fgm", "plackett"),
  x,
  param = 0.5,
  param.est = TRUE,
  df = 4,
  df.est = TRUE,
  margins = "ranks",
  flip = 0,
  M = 1000,
  dispstr = "ex",
  lower = NULL,
  upper = NULL,
  seed.active = NULL,
  processes = 1
)
```

Arguments

copula	The copula to test for. Possible are "normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos", "fgm" and "plackett".
x	A matrix containing the data with rows being observations and columns being variables.
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 for the margins shall be used. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively the following distributions can be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp"). Input can be either one method, e.g. "ranks", which will be used for estimation of all data sequences. Also an individual method for each margin can be specified, e.g. c("ranks", "norm", "t") for 3 data sequences. If one does not want to estimate the margins, set it to NULL.

flip	The control parameter to flip the copula by 90, 180, 270 degrees clockwise. Only applicable for bivariate copula. Default is 0 and possible inputs are 0, 90, 180, 270 and NULL.
M	Number of bootstrapping loops.
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.
lower	Lower bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
upper	Upper bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
seed.active	Has to be either an integer or a vector of M+1 integers. If an integer, then the seeds for the bootstrapping procedure will be simulated. If M+1 seeds are provided, then these seeds are used in the bootstrapping procedure. Defaults to NULL, then R generates the seeds from the computer runtime. Controlling the seeds is useful for reproducibility of a simulation study to compare the power of the tests or for reproducibility of an empirical study.
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 Anderson-Darling test statistic (supposedly) computes U[0,1]-distributed (under H_0) random variates via the distribution function of chi-square distribution with d degrees of freedom, see Hofert et al. (2014). 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 $\mathcal{R} : (0,1)^d \to (0,1)^d$. 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

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

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

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

The Anderson-Darling test statistic of the variates

$$G(x_j) = \chi^2_d(x_j)$$

is computed (via `ADGofTest::ad.test`), where $x_j = \sum_{i=1}^{d} (\Phi^{-1}(e_{ij}))^2$, Φ^{-1} denotes the quantile function of the standard normal distribution function, χ^2_d denotes the distribution function of the chi-square distribution with `d` degrees of freedom, and u_{ij} is the jth component in the ith 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 statistc, respectively.

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

Value

An object of the `class` gofCOP with the components

`method`	a character which informs about the performed analysis
`copula`	the copula tested for
`margins`	the method used to estimate the margin distribution.
`param.margins`	the parameters of the estimated margin distributions. Only applicable if the margins were not specified as `"ranks"` or `NULL`.
`theta`	dependence parameters of the copulae
`df`	the degrees of freedem of the copula. Only applicable for t-copula.
`res.tests`	a matrix with the p-values and test statistics of the hybrid and the individual tests

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.* doi: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(IndexReturns2D)

gofRosenblattChisq("normal", IndexReturns2D, M = 10)
```

`gofRosenblattGamma` *Gof test using the Anderson-Darling test statistic and the gamma distribution*

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 inbuild 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"`, `"clayton"`, `"gumbel"`, `"frank"`, `"joe"`, `"amh"`, `"galambos"`, `"fgm"` and `"plackett"`. 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 = c("normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos",
    "fgm", "plackett"),
  x,
  param = 0.5,
  param.est = TRUE,
  df = 4,
  df.est = TRUE,
  margins = "ranks",
  flip = 0,
  M = 1000,
  dispstr = "ex",
  lower = NULL,
  upper = NULL,
  seed.active = NULL,
  processes = 1
)
```

Arguments

`copula`	The copula to test for. Possible are `"normal"`, `"t"`, `"clayton"`, `"gumbel"`, `"frank"`, `"joe"`, `"amh"`, `"galambos"`, `"fgm"` and `"plackett"`.

x	A matrix containing the data with rows being observations and columns being variables.
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 for the margins shall be used. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively the following distributions can be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp"). Input can be either one method, e.g. "ranks", which will be used for estimation of all data sequences. Also an individual method for each margin can be specified, e.g. c("ranks", "norm", "t") for 3 data sequences. If one does not want to estimate the margins, set it to NULL.
flip	The control parameter to flip the copula by 90, 180, 270 degrees clockwise. Only applicable for bivariate copula. Default is 0 and possible inputs are 0, 90, 180, 270 and NULL.
M	Number of bootstrapping loops.
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.
lower	Lower bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
upper	Upper bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
seed.active	Has to be either an integer or a vector of M+1 integers. If an integer, then the seeds for the bootstrapping procedure will be simulated. If M+1 seeds are provided, then these seeds are used in the bootstrapping procedure. Defaults to NULL, then R generates the seeds from the computer runtime. Controlling the seeds is useful for reproducibility of a simulation study to compare the power of the tests or for reproducibility of an empirical study.
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 Anderson-Darling test statistic (supposedly) computes U[0,1]-distributed (under H_0) random variates via the distribution function of the gamma distribution, see Hofert et al. (2014). 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 $\mathcal{R} : (0,1)^d \to (0,1)^d$. 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

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

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

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

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 statistc, respectively.

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

Value

An object of the `class` gofCOP with the components

`method`	a character which informs about the performed analysis
`copula`	the copula tested for
`margins`	the method used to estimate the margin distribution.
`param.margins`	the parameters of the estimated margin distributions. Only applicable if the margins were not specified as `"ranks"` or `NULL`.
`theta`	dependence parameters of the copulae
`df`	the degrees of freedem of the copula. Only applicable for t-copula.
`res.tests`	a matrix with the p-values and test statistics of the hybrid and the individual tests

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*. doi: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(IndexReturns2D)

gofRosenblattGamma("normal", IndexReturns2D, M = 10)
```

`gofRosenblattSnB` *The SnB test based on the Rosenblatt transformation*

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"`, `"clayton"`, `"gumbel"`, `"frank"`, `"joe"`, `"amh"`, `"galambos"`, `"fgm"` and `"plackett"`. 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 = c("normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos",
    "fgm", "plackett"),
 x,
 param = 0.5,
 param.est = TRUE,
 df = 4,
 df.est = TRUE,
 margins = "ranks",
 flip = 0,
 M = 1000,
 dispstr = "ex",
```

```
  lower = NULL,
  upper = NULL,
  seed.active = NULL,
  processes = 1
)
```

Arguments

copula	The copula to test for. Possible are "normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos", "fgm" and "plackett".
x	A matrix containing the data with rows being observations and columns being variables.
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 for the margins shall be used. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively the following distributions can be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp"). Input can be either one method, e.g. "ranks", which will be used for estimation of all data sequences. Also an individual method for each margin can be specified, e.g. c("ranks", "norm", "t") for 3 data sequences. If one does not want to estimate the margins, set it to NULL.
flip	The control parameter to flip the copula by 90, 180, 270 degrees clockwise. Only applicable for bivariate copula. Default is 0 and possible inputs are 0, 90, 180, 270 and NULL.
M	Number of bootstrapping loops.
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.
lower	Lower bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
upper	Upper bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
seed.active	Has to be either an integer or a vector of M+1 integers. If an integer, then the seeds for the bootstrapping procedure will be simulated. If M+1 seeds are provided, then these seeds are used in the bootstrapping procedure. Defaults to NULL, then R generates the seeds from the computer runtime. Controlling the seeds is useful for reproducibility of a simulation study to compare the power of the tests or for reproducibility of an empirical study.

`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 $\mathcal{R} : (0,1)^d \to (0,1)^d$ 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

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

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

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

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

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 statistc, respectively.

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

Value

An object of the `class` gofCOP with the components

`method`	a character which informs about the performed analysis
`copula`	the copula tested for
`margins`	the method used to estimate the margin distribution.
`param.margins`	the parameters of the estimated margin distributions. Only applicable if the margins were not specified as `"ranks"` or `NULL`.
`theta`	dependence parameters of the copulae
`df`	the degrees of freedem of the copula. Only applicable for t-copula.
`res.tests`	a matrix with the p-values and test statistics of the hybrid and the individual tests

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*. doi: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(IndexReturns2D)

gofRosenblattSnB("normal", IndexReturns2D, 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"`, `"clayton"`, `"gumbel"`, `"frank"`, `"joe"`, `"amh"`, `"galambos"`, `"fgm"` and `"plackett"`. 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 = c("normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos",
    "fgm", "plackett"),
 x,
 param = 0.5,
 param.est = TRUE,
 df = 4,
 df.est = TRUE,
 margins = "ranks",
 flip = 0,
 M = 1000,
 dispstr = "ex",
```

```
  lower = NULL,
  upper = NULL,
  seed.active = NULL,
  processes = 1
)
```

Arguments

copula	The copula to test for. Possible are "normal", "t", "clayton", "gumbel", "frank", "joe", "amh", "galambos", "fgm" and "plackett".
x	A matrix containing the data with rows being observations and columns being variables.
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 for the margins shall be used. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively the following distributions can be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp"). Input can be either one method, e.g. "ranks", which will be used for estimation of all data sequences. Also an individual method for each margin can be specified, e.g. c("ranks", "norm", "t") for 3 data sequences. If one does not want to estimate the margins, set it to NULL.
flip	The control parameter to flip the copula by 90, 180, 270 degrees clockwise. Only applicable for bivariate copula. Default is 0 and possible inputs are 0, 90, 180, 270 and NULL.
M	Number of bootstrapping loops.
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.
lower	Lower bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
upper	Upper bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
seed.active	Has to be either an integer or a vector of M+1 integers. If an integer, then the seeds for the bootstrapping procedure will be simulated. If M+1 seeds are provided, then these seeds are used in the bootstrapping procedure. Defaults to NULL, then R generates the seeds from the computer runtime. Controlling the seeds is useful for reproducibility of a simulation study to compare the power of the tests or for reproducibility of an empirical study.

`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 $\mathcal{R} : (0,1)^d \to (0,1)^d$ 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

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

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

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

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

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 statistc, respectively.

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

Value

An object of the `class` gofCOP with the components

`method`	a character which informs about the performed analysis
`copula`	the copula tested for
`margins`	the method used to estimate the margin distribution.
`param.margins`	the parameters of the estimated margin distributions. Only applicable if the margins were not specified as `"ranks"` or `NULL`.
`theta`	dependence parameters of the copulae
`df`	the degrees of freedem of the copula. Only applicable for t-copula.
`res.tests`	a matrix with the p-values and test statistics of the hybrid and the individual tests

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*. doi: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(IndexReturns2D)

gofRosenblattSnC("normal", IndexReturns2D, M = 10)
```

`gofTest4Copula` *Applicable gof tests for testing problem*

Description

`gofTest4Copula` returns for a given copula and a dimension the applicable implemented tests.

Usage

```
gofTest4Copula(copula = NULL, d = 2)
```

Arguments

`copula`	The copula to test for. Possible are the copulae `"normal"`, `"t"`, `"clayton"`, `"gumbel"`, `"frank"`, `"joe"`, `"amh"`, `"galambos"`, `"huslerReiss"`, `"tawn"`, `"tev"`, `"fgm"` and `"plackett"`. For the default `NULL`, all tests in the package are returned.
`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

```
gofTest4Copula("clayton", d = 2)

gofTest4Copula("gumbel", d = 5)
```

`gofWhich` *The function* `gofWhich` *was renamed to* `gofTest4Copula`. *Please use* `gofTest4Copula`.

Description

Please see and use the function `gofTest4Copula`.

Usage

```
gofWhich(copula = NULL, d = 2)
```

Arguments

`copula`	The copula to test for. Possible are the copulae `"normal"`, `"t"`, `"clayton"`, `"gumbel"`, `"frank"`, `"joe"`, `"amh"`, `"galambos"`, `"huslerReiss"`, `"tawn"`, `"tev"`, `"fgm"` and `"plackett"`. For the default `NULL`, all tests in the package are returned.
`d`	The dimension to search for.

Value

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

`gofWhichCopula` *The function* `gofWhichCopula` *was renamed to* `gofCopula4Test`. *Please use* `gofCopula4Test`.

Description

Please see and use the function `gofCopula4Test`.

Usage

```
gofWhichCopula(test)
```

Arguments

`test`	The test to search for copula.

Value

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

`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"`, `"clayton"`, `"gumbel"`, `"frank"` and `"joe"`. 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 = c("normal", "t", "clayton", "gumbel", "frank", "joe"),
  x,
  param = 0.5,
  param.est = TRUE,
  df = 4,
  df.est = TRUE,
  margins = "ranks",
  flip = 0,
  M = 1000,
  lower = NULL,
  upper = NULL,
  seed.active = NULL,
  processes = 1
)
```

Arguments

`copula`	The copula to test for. Possible are the copulae `"normal"`, `"t"`, `"clayton"`, `"gumbel"`, `"frank"` and `"joe"`.
`x`	A matrix containing the data with rows being observations and columns being variables.
`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	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.
margins	Specifies which estimation method for the margins shall be used. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively the following distributions can be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp"). Input can be either one method, e.g. "ranks", which will be used for estimation of all data sequences. Also an individual method for each margin can be specified, e.g. c("ranks", "norm", "t") for 3 data sequences. If one does not want to estimate the margins, set it to NULL.
flip	The control parameter to flip the copula by 90, 180, 270 degrees clockwise. Only applicable for bivariate copula. Default is 0 and possible inputs are 0, 90, 180, 270 and NULL.
M	Number of bootstrap samples.
lower	Lower bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
upper	Upper bound for the maximum likelihood estimation of the copula parameter. The constraint is also active in the bootstrapping procedure. The constraint is not active when a switch to inversion of Kendall's tau is necessary. Default NULL.
seed.active	Has to be either an integer or a vector of M+1 integers. If an integer, then the seeds for the bootstrapping procedure will be simulated. If M+1 seeds are provided, then these seeds are used in the bootstrapping procedure. Defaults to NULL, then R generates the seeds from the computer runtime. Controlling the seeds is useful for reproducibility of a simulation study to compare the power of the tests or for reproducibility of an empirical study.
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 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

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

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

The test statistic is derived by

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

with

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

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

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

and

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

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

The test statistic will be rejected if

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

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

Please note, the test gofWhite may be unstable for t-copula. Please handle the results carefully.

Value

An object of the `class` gofCOP with the components

`method`	a character which informs about the performed analysis
`copula`	the copula tested for
`margins`	the method used to estimate the margin distribution.
`param.margins`	the parameters of the estimated margin distributions. Only applicable if the margins were not specified as `"ranks"` or `NULL`.
`theta`	dependence parameters of the copulae
`df`	the degrees of freedem of the copula. Only applicable for t-copula.
`res.tests`	a matrix with the p-values and test statistics of the hybrid and the individual tests

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.* https://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(IndexReturns2D)

gofWhite("normal", IndexReturns2D, M = 10)
```

IndexReturns2D *Log returns of european stock indices*

Description

A dataset containing the log returns of two european stock indices in the year 1998.

Format

A matrix with 100 rows and 2 columns

Source

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

IndexReturns3D *Log returns of european stock indices*

Description

A dataset containing the log returns of three european stock indices in the year 1998.

Format

A matrix with 200 rows and 3 columns

Source

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

plot.gofCOP *Plotting function for objects of class gofCOP*

Description

Plots an object of class gofCOP.

Usage

```
## S3 method for class 'gofCOP'
plot(
  x,
  copula = NULL,
  hybrid = NULL,
  point.bg = "white",
  point.col = "black",
  point.cex = 0.7,
  jitter.val = 0.1,
  pal = "xmen",
  bean.b.o = 0.2,
  inf.method = "hdi",
  theme = 2,
  ...
)
```

Arguments

x	An object of class gofCOP.
copula	Specify for which copulae to plot the results. The possible entry is a character vector of the copula names. Default is NULL which plots all copulae in the object.
hybrid	Specify for which combinations to plot the hybrid tests. The possible entry is a numeric vector of the hybrid combinations. Default is NULL which plots all possible combinations.
point.bg	The background color of the plot. Defaults to white.
point.col	The color of the dots. Defaults to black.
point.cex	The size of the dots. Defaults to 0.7.
jitter.val	Jittering the dots left/right for better visibility of results. Defaults to 0.1.
pal	Color palette of the plots. Defaults to xmen.
bean.b.o	Specification of how opaque the bars are. Defaults to 0.2.
inf.method	Type of inference bands. Defaults to Bayesian Highest Density Intervals, hdi.
theme	Theme of the plot. See pirateplot for more details. Defaults to 2.
...	Further arguments to be passed to pirateplot.

Details

The plotting function is constructed around pirateplot from the yarrr package. Please see respective package for more details on the non-default specifications of the plotting function.

We recommend not to amend the arguments xlim, data, formula, sortx, xaxt, ylim and ylab from pirateplot. The arguments were defined such that the resulting plot displays the test results in a proper manner.

Value

None

References

Phillips, N. (2017). yarrr: A Companion to the e-Book "YaRrr!: The Pirate's Guide to R". *R package version 0.1.5.* https://CRAN.R-project.org/package=yarrr

---

print.gofCOP *Printing function for objects of class gofCOP*

Description

print.gofCOP prints the values of an object of class gofCOP.

Usage

```
## S3 method for class 'gofCOP'
print(x, ...)
```

Arguments

x	An object of class gofCOP.
...	Further arguments to be passed over. Currently non.

Value

None

print.goftime *Printing function for an object of class goftime. goftime objects are created by the function gofCheckTime.*

Description

Printing function for an object of class goftime. goftime objects are created by the function gofCheck-Time.

Usage

```
## S3 method for class 'goftime'
print(x, ...)
```

Arguments

x	An object of class goftime
...	Additional arguments passed to print

Index

Banks, 3

CopulaTestTable, 3, *3*
CryptoCurrencies, 4

gof, 4, *4*
gofArchmChisq, 7, *7*
gofArchmGamma, 10, *11*
gofArchmSnB, 14, *14*
gofArchmSnC, 17, *17*
gofCheckTime, 20, *20*
gofco, 22, *22*
gofCopula4Test, 25, *25*, *66*
gofCustomTest, 25, *25*
gofCvM, 28, *28*
gofGetHybrid, 31, *31*
gofKendallCvM, 32, *32*
gofKendallKS, 35, *35*
gofKernel, 38
gofKS, 42, *42*
gofOutputHybrid, 45, *45*
gofPIOSRn, 46
gofPIOSTn, 49
gofRosenblattChisq, 52, *52*
gofRosenblattGamma, 56, *56*
gofRosenblattSnB, 59, *59*
gofRosenblattSnC, 62, *62*
gofTest4Copula, 65, *65, 66*
gofTstat, *7*, *11*, *14*, *17*, *28*, *42*, *53*, *56*, *59*, *62*
gofWhich, 66
gofWhichCopula, 66
gofWhite, 67, *67*

IndexReturns2D, 70
IndexReturns3D, 70

pirateplot, *71, 72*
plot.gofCOP, 71
print.gofCOP, 72, *72*
print.goftime, 73