Package ‘gofCopula’

November 4, 2015

Type Package

Title Goodness-of-Fit Tests for Copulae

Version 0.1-1

Date 2015-10-27

Author
Ostap Okhrin <ostap.okhrin@tu-dresden.de>, Shulin Zhang <slzhang@swufe.edu.cn>, Qian M. Zhou <qmzhou@sfu.c
mon Trimborn <simon.trimborn@wiwi.hu-berlin.de>

Maintainer Simon Trimborn <simon.trimborn@wiwi.hu-berlin.de>

LazyLoad yes

Depends R (>= 1.9.0), copula

Imports SparseGrid, numDeriv, VineCopula, methods, stats

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

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-11-04 19:04:37

R topics documented:

`gof` *Combining function for tests*

Description

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

Usage

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

Arguments

`x`	A matrix containing the residuals of the data.
`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. This are e.g. `"gaussian"` and `"clayton"`. If `"copula"` is chosen, the tests which are able to test for `"gaussian"`, `"t"`, `"frank"`, `"gumbel"` and `"clayton"` are performed. If one of the arguments `tests` or `copula` is not `NULL`, then `priority` doesn't affect the choice of the copulae and tests.
`copula`	A character vector which indicates the copula to test for.
`tests`	A character vector which indicates the tests to use.
`M`	The amount of bootstrap rounds to be performed by each test. Default is 1000.
`MJ`	Just for the test gofKernel. Size of bootstrapping sample.
`param`	The copulae parameters to use for each test, if it shall not be estimated.
`param.est`	Shall be either `TRUE` or `FALSE`. `TRUE` means that `param` will be estimated.
`df`	The degrees of freedom, if not meant to be estimated. Only necessary if tested for `"t"`-copula.
`df.est`	Indicates if `df` shall be estimated. Has to be either `FALSE` or `TRUE`, where `TRUE` means that it will be estimated.

m	Length of blocks. Only necessary if the test gofPIOSTn is part of testset.
delta.J	Scaling parameter for the matrix of smoothing parameters. Only necessary if the test gofKernel is part of testset.
nodes.Integration	Number of knots of the bivariate Gauss-Legendre quadrature. Only necessary if the test gofKernel is part of testset.
m_b	The power of the statistic. Only necessary if the test gofRn is part of testset.
zeta.m	The adjustment parameter. Only necessary if the test gofRn is part of testset.
b_Rn	The bandwidth for the estimation of the first-order partial derivatives based on the empirical copula. Only necessary if the test gofRn is part of testset.

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 will be this tests performed for all implemented copulae. If character vectors are given for copula and tests, then the tests are performed with the given copulae. If tests = NULL and copula = NULL, then the argument priority catches in and defines the procedure.

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

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
erg.tests	a matrix with the p-values and test statistics of the individual tests including the hybrid test

Examples

```
data = cbind(rnorm(100), rnorm(100), rnorm(100))

gof(data, priority = "tests", copula = "gaussian",
tests = c("gofRosenblattSnB", "gofRosenblattSnC"), M = 20)
```

---

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

Description

gofADChisq is a wrapper for the functions gofCopula, fitCopula, ellipCopula and archmCopula from the package **copula**. It combines these functions to test a dataset for a copula directly without all the necessary intermediate steps and extends its functionality. The margins can be estimated by a bunch of distributions and the time which is necessary for the estimation can be given. gofADChisq contains the ADChisq 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 approximate p-values are computed with a parametric bootstrap. It is possible to insert datasets of all dimensions above 1 and the possible copulae are "gaussian", "t", "gumbel", "clayton" and "frank". The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used.

Usage

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

Arguments

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

Details

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

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

with $\mathcal{C}_0$ as the true class of copulae under H_0. The Anderson-Darling test statistic of the variates

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

is computed (via `ADGofTest::ad.test`), where $x_i = \sum_{j=1}^{d} (\Phi^{-1}(u_{ij}))^2$, 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

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

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

$$(0.5 + \sum_{b=1}^{M} 1_{\{T_b \leq T\}} / (M+1),$$

where T and T_b denote the test statistic and the bootstrapped test statistc, respectively. This ensures that the approximate p-value is a number strictly between 0 and 1, which is sometimes necessary for further treatments. See Pesarin (2001) for more details.

Value

A object of the `class` gofCOP with the components

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

References

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

Marius Hofert, Ivan Kojadinovic, Martin Maechler, Jun Yan (2014). copula: Multivariate Dependence with Copulas. *R package version 0.999-12.*. http://CRAN.R-project.org/package=copula

Pesarin, F. (2001). *Multivariate Permutation Tests: With applications in Biostatistics*, Wiley

Examples

```
data = cbind(rnorm(100), rnorm(100), rnorm(100))

gofADChisq("gaussian", data, M = 20)
```

gofADGamma — *Gof test using the Anderson-Darling test statistic and the gamma distribution*

Description

gofADGamma is a wrapper for the functions gofCopula, fitCopula, ellipCopula and archmCopula from the package **copula**. It combines these functions to test a dataset for a copula directly without all the necessary intermediate steps and extends its functionality. The margins can be estimated by a bunch of distributions and the time which is necessary for the estimation can be given. gofADGamma contains the ADGamma 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 approximate p-values are computed with a parametric bootstrap. It is possible to insert datasets of all dimensions above 1 and the possible copulae are "gaussian", "t", "gumbel", "clayton" and "frank". The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used.

Usage

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

Arguments

copula — The copula to test for. Possible are "gaussian", "t", "clayton" and "gumbel".

x — A matrix containing the residuals of the data.

M — Number of bootstrapping loops.

param — The copula parameter to use, if it shall not be estimated.

param.est — Shall be either TRUE or FALSE. TRUE means that param will be estimated.

df — Degrees of freedom, if not meant to be estimated. Only necessary if tested for "t"-copula.

df.est — Indicates if df shall be estimated. Has to be either FALSE or TRUE, where TRUE means that it will be estimated.

margins — Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp").

execute.times.comp — Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if M is at least 100.

Details

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

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

with $\mathcal{C}_0$ as the true class of copulae under H_0. The Anderson-Darling test statistic of the variates

$$G(x_i) = \Gamma_d\left(x_i\right)$$

is computed (via `ADGofTest::ad.test`), where $x_i = \sum_{j=1}^{d}(-\ln u_{ij})$, $\Gamma_d()$ 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

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

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

$$(0.5 + \sum_{b=1}^{N} \mathbf{1}_{\{T_b \leq T\}}/(N+1),$$

where T and T_b denote the test statistic and the bootstrapped test statistc, respectively. This ensures that the approximate p-value is a number strictly between 0 and 1, which is sometimes necessary for further treatments. See Pesarin (2001) for more details.

Value

A object of the `class` gofCOP with the components

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

References

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

Marius Hofert, Ivan Kojadinovic, Martin Maechler, Jun Yan (2014). copula: Multivariate Dependence with Copulas. *R package version 0.999-12.*. http://CRAN.R-project.org/package=copula

Pesarin, F. (2001). *Multivariate Permutation Tests: With applications in Biostatistics*, Wiley

Examples

```
data = cbind(rnorm(100), rnorm(100), rnorm(100))

gofADGamma("gaussian", data, M = 20)
```

---

gofCheckTime *Combining function for tests*

Description

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

Usage

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

Arguments

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

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

m_b
: The power of the statistic. Only necessary if the test gofRn is part of testset.

zeta.m
: The adjustment parameter. Only necessary if the test gofRn is part of testset.

b_Rn
: The bandwidth for the estimation of the first-order partial derivatives based on the empirical copula. Only necessary if the test gofRn is part of testset.

Details

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

Examples

```
data = cbind(rnorm(100), rnorm(100), rnorm(100))

gofCheckTime("gaussian", data, "gofRosenblattSnB", M = 10000)
```

---

gofco *Interface with copula class*

Description

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

Usage

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

Arguments

copulaobject
: An object with the copula class from the copula package.

x
: A matrix containing the residuals of the data. Take into account that the dimension of the matrix must fit the required dimensions from the single tests.

testset
: The tests to be used. Possible are "gofPIOSRn", "gofPIOSTn", "gofKernel", "gofRosenblattSnB", "gofRosenblattSnC", "gofADChisq", "gofADGamma", "gofSn", "gofKendallCvM", "gofKendallKS", "gofWhite", "gofRn".

margins
: Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp").

M
: Number of bootstrapping samples in the single tests.

execute.times.comp
: Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if `M` is at least 100.

m
: Length of blocks. Only necessary if the test `gofPIOSTn` is part of `testset`.

MJ
: Size of bootstrapping sample. Only necessary if the test `gofKernel` is part of `testset`.

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

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

m_b
: The power of the statistic. Only necessary if the test `gofRn` is part of `testset`.

zeta.m
: The adjustment parameter. Only necessary if the test `gofRn` is part of `testset`.

b_Rn
: The bandwidth for the estimation of the first-order partial derivatives based on the empirical copula. Only necessary if the test `gofRn` is part of `testset`.

Details

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

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

Value

A object of the `class` gofCOP with the components

method
: a character which informs about the performed analysis

erg.tests
: a matrix with the p-values and test statistics of the hybrid and the individual tests

References

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

Examples

```
data = cbind(rnorm(100), rnorm(100))
copObject = normalCopula()

gofco(copObject, x = data, testset = c("gofPIOSRn", "gofKernel"), M = 20)
```

gofHybrid *Hybrid gof test*

Description

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

Usage

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

Arguments

copula	The copula to test for. Possible are "gaussian", "t", "clayton", "gumbel" and "frank".
x	A matrix containing the residuals of the data. Take into account that the dimension of the matrix must fit the required dimensions from the single tests.
testset	The tests to be used. Possible are "gofPIOSRn", "gofPIOSTn", "gofKernel", "gofRosenblattSnB", "gofRosenblattSnC", "gofADChisq", "gofADGamma", "gofSn", "gofKendallCvM", "gofKendallKS", "gofWhite", "gofRn".
margins	Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp").
M	Number of bootstrapping samples in the single tests.
execute.times.comp	Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if M is at least 100.
param	The copulae parameters to use for each test, if it shall not be estimated.
param.est	Shall be either TRUE or FALSE. TRUE means that param will be estimated.
df	The degrees of freedom, if not meant to be estimated. Only necessary if tested for "t"-copula.
df.est	Indicates if df shall be estimated. Has to be either FALSE or TRUE, where TRUE means that it will be estimated.
m	Length of blocks. Only necessary if the test gofPIOSTn is part of testset.
MJ	Size of bootstrapping sample. Only necessary if the test gofKernel is part of testset.

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

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

m_b — The power of the statistic. Only necessary if the test `gofRn` is part of `testset`.

zeta.m — The adjustment parameter. Only necessary if the test `gofRn` is part of `testset`.

b_Rn — The bandwidth for the estimation of the first-order partial derivatives based on the empirical copula. Only necessary if the test `gofRn` is part of `testset`.

Details

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

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

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

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

A object of the `class` gofCOP with the components

method — a character which informs about the performed analysis

erg.tests — a matrix with the p-values and test statistics of the hybrid and the individual tests

References

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

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

Examples

```
data = cbind(rnorm(100), rnorm(100))

gofHybrid("gaussian", data, testset = c("gofRosenblattSnB", "gofRosenblattSnC"), M = 10)
```

gofKendallCvM — *2 dimensional gof test (Cramer-von Mises) based on Kendall's process*

Description

gofKendallCvM is a wrapper for the function BiCopGofTest from the package **VineCopula** which extends its functionality. The margins can be estimated by a bunch of distributions and the time which is necessary for the estimation can be given. It tests a given 2 dimensional dataset for a copula based on Kendall's process with the Cramer-von Mises test statistic. The possible copulae are "gaussian", "gumbel", "clayton" and "frank". See for reference Genest et al. (2009). The parameter estimation is performed with pseudo maximum likelihood method.

Usage

```
gofKendallCvM(copula, x, param = 0, param.est = T, margins = "ranks",
              M = 100, execute.times.comp = T)
```

Arguments

- `copula` The copula to test for. Possible are the copulae "gaussian", "clayton", "gumbel" and "frank".
- `x` A 2 dimensional matrix containing the residuals of the data.
- `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.
- `margins` Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp").
- `M` Number of bootstrap samples.
- `execute.times.comp` Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if M is at least 100.

Details

With the pseudo observations U_{ij} for $i = 1, \dots, n$, $j = 1, \dots, d$ and $\mathbf{u} \in [0,1]^d$ is the empirical copula given by $C_n(\mathbf{u}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{I}(U_{i1} \le u_1, \dots, U_{id} \le u_d)$. Let the rescaled pseudo observations be $\mathbf{V}_1 = C_n(\mathbf{U}_1), \dots, \mathbf{V}_n = C_n(\mathbf{U}_n)$ and the distribution function of $\mathbf{V}$ shall be K. The estimated version is given by

$$K_n(v) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{I}(\mathbf{V}_i \le v)$$

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 the H_0 is the test not necessarily consistent.

The approximate p-value is computed by the formula

$$\frac{1}{M} \sum_{b=1}^{M} \mathbf{I}_{\{T_b \leq T\}}.$$

Value

A object of the `class` gofCOP with the components

`method` a character which informs about the performed analysis

`erg.tests` a matrix with the p-value and test statistic of test

References

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

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 = cbind(rnorm(100), rnorm(100))

gofKendallCvM("gaussian", data, M = 1)
```

`gofKendallKS` *2 dimensional gof test (Kolmogorov-Smirnof) based on Kendall's process*

Description

gofKendallKS is a wrapper for the function BiCopGofTest from the package **VineCopula** which extends its functionality. The margins can be estimated by a bunch of distributions and the time which is necessary for the estimation can be given. It tests a given 2 dimensional dataset for a copula based on Kendall's process with the Kolmogorov-Smirnof test statistic. The possible copulae are "gaussian", "gumbel", "clayton" and "frank". See for reference Genest Genest et al. (2009). The parameter estimation is performed with pseudo maximum likelihood method.

Usage

```
gofKendallKS(copula, x, param = 0, param.est = T, margins = "ranks",
               M = 100, execute.times.comp = T)
```

Arguments

copula	The copula to test for. Possible are the copulae "gaussian", "clayton", "gumbel" and "frank".
x	A 2 dimensional matrix containing the residuals of the data.
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.
margins	Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp").
M	Number of bootstrap samples.
execute.times.comp	Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if M is at least 100.

Details

With the pseudo observations U_{ij} for $i = 1, \dots, n$, $j = 1, \dots, d$ and $\mathbf{u} \in [0, 1]^d$ is the empirical copula given by $C_n(\mathbf{u}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{I}(U_{i1} \le u_1, \dots, U_{id} \le u_d)$. Let the rescaled pseudo observations be $\mathbf{V}_1 = C_n(\mathbf{U}_1), \dots, \mathbf{V}_n = C_n(\mathbf{U}_n)$ and the distribution function of $\mathbf{V}$ shall be K. The estimated version is given by

$$K_n(v) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{I}(\mathbf{V}_i \le v)$$

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 $p \ge 1$, see Genest et al. (2009). The resulting Cramer-von Mises test statistic is then given by

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

Because $H_0^{'}$ consists of more distributions than the H_0 is the test not necessarily consistent.

The approximate p-value is computed by the formula

$$\frac{1}{M}\sum_{b=1}^{M}\mathbf{I}_{\{T_b\leq T\}}.$$

Value

A object of the `class` gofCOP with the components

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

References

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

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 = cbind(rnorm(100), rnorm(100))

gofKendallKS("gaussian", data, M = 1)
```

`gofKernel` *2 dimensional gof test of Scaillet*

Description

`gofKernel` tests a 2 dimensional dataset with the Scaillet test for a copula. The possible copulae are "gaussian", "t", "gumbel", "clayton" and "frank". The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used.

Usage

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

Arguments

- `copula` — The copula to test for. Possible are the copulae "gaussian", "t", "clayton", "gumbel" and "frank".
- `x` — A 2 dimensional matrix containing the residuals of the data.
- `M` — Number of bootstrapping loops.
- `param` — The parameter to be used.
- `param.est` — Shall be either `TRUE` or `FALSE`. `TRUE` means that `param` will be estimated with a maximum likelihood estimation.
- `df` — Degrees of freedom, if not meant to be estimated. Only necessary if tested for "t"-copula.
- `df.est` — Indicates if `df` shall be estimated. Has to be either `FALSE` or `TRUE`, where `TRUE` means that it will be estimated.
- `margins` — Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp").
- `MJ` — Size of bootstrapping sample.
- `delta.J` — Scaling parameter for the matrix of smoothing parameters.
- `nodes.Integration` — Number of knots of the bivariate Gauss-Legendre quadrature.
- `execute.times.comp` — Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if `M` is at least 100.

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 $H = 2.6073n^{-1/6}\hat{\Sigma}^{1/2}$ with $\hat{\Sigma}$ the sample covariance matrix. The test statistic is then given by

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

where $*$ denotes the convolution operator and 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

$$p = \frac{1}{M}\sum_{b=1}^{M} \mathbf{I}(|T_b| \geq T).$$

Value

A object of the `class` gofCOP with the components

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

References

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

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

Examples

```
data = cbind(rnorm(100), rnorm(100))

gofKernel("gaussian", data, M = 10)
```

---

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

Description

`gofPIOSRn` tests a 2 dimensional dataset with the approximate PIOS test for a copula. The possible copulae are "gaussian", "t", "gumbel", "clayton" and "frank". The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used.

Usage

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

Arguments

`copula`	The copula to test for. Possible are the copulae `"gaussian"`, `"t"`, `"clayton"`, `"gumbel"` and `"frank"`.
`x`	A 2 dimensional matrix containing the residuals of the data.
`M`	Number of bootstrapping loops.
`param`	The parameter to be used.
`param.est`	Shall be either `TRUE` or `FALSE`. `TRUE` means that `param` will be estimated with a maximum likelihood estimation.

`df` Degrees of freedom, if not meant to be estimated. Only necessary if tested for `"t"`-copula.

`df.est` Indicates if `df` shall be estimated. Has to be either `FALSE` or `TRUE`, where `TRUE` means that it will be estimated.

`margins` Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is `"ranks"`, which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: `"beta"`, `"cauchy"`, Chi-squared (`"chisq"`), `"f"`, `"gamma"`, Log normal (`"lnorm"`), Normal (`"norm"`), `"t"`, `"weibull"`, Exponential (`"exp"`).

`execute.times.comp` Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if `M` is at least 100.

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

$$T = \sum_{b=1}^{B} \sum_{i=1}^{m} [l\{U_i^b; \theta_n\} - l\{U_i^b; \theta_n^{-b}\}]$$

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

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

and

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

By defining two information matrices

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

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

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

$$p = \frac{1}{B} \sum_{b=1}^{B} \mathbf{I}(|T_b| \geq T).$$

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).

Value

A object of the `class` gofCOP with the components

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

References

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

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

Examples

```
data = cbind(rnorm(100), rnorm(100))

gofPIOSRn("gaussian", data, M = 20)
```

---

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

Description

gofPIOSTn tests a 2 dimensional dataset with the PIOS test for a copula. The possible copulae are "gaussian", "t", "gumbel", "clayton" and "frank". The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used.

Usage

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

Arguments

copula	The copula to test for. Possible are the copulae "gaussian", "t", "clayton", "gumbel" and "frank".
x	A 2 dimensional matrix containing the residuals of the data.
M	Number of bootstrapping loops.
param	The parameter to be used.
param.est	Shall be either TRUE or FALSE. TRUE means that param will be estimated with a maximum likelihood estimation.
df	Degrees of freedom, if not meant to be estimated. Only necessary if tested for "t"-copula.
df.est	Indicates if df shall be estimated. Has to be either FALSE or TRUE, where TRUE means that it will be estimated.
margins	Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp").
m	Length of blocks.
execute.times.comp	Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if M is at least 100.

Details

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

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

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

$$T = \sum_{b=1}^{B} \sum_{i=1}^{m} [l\{U_i^b; \theta_n\} - l\{U_i^b; \theta_n^{-b}\}]$$

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

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

and

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

The approximate p-value is computed by the formula

$$p = \frac{1}{M}\sum_{b=1}^{M}\mathbf{I}(|T_b| \geq T).$$

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

Value

A object of the `class` gofCOP with the components

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

References

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

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

Examples

```
data = cbind(rnorm(100), rnorm(100))

gofPIOSTn("gaussian", data, M = 20)
```

---

gofRn — *The Rn gof test from package* **copula**

Description

`gofRn` is a wrapper for the functions `gofCopula`, `fitCopula`, `ellipCopula` and `archmCopula` from the package **copula**. It combines these functions to test a dataset for a copula directly without all the necessary intermediate steps. `gofRn` performs the gof test from Genest et al. (2013) for copulae and compares the empirical copula against a parametric estimate of the copula derived under the null hypothesis. The approximate p-values are computed with a fast multiplier approach. It is just possible to insert datasets of dimension 2 and the possible copulae are "gaussian", "t", "gumbel", "clayton" and "frank". The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used.

Usage

```
gofRn(copula, x, M = 1000, param = 0.5, param.est = T, df = 4, df.est = T,
      m_b = 0.5, zeta.m = 0, b_Rn = 0.05, execute.times.comp = T)
```

Arguments

`copula`	The copula to test for. Possible are "gaussian", "t", "clayton", "gumbel" and "frank".
`x`	A matrix containing the residuals of the data.
`M`	Number of bootstrapping loops.
`param`	The copula parameter to use, if it shall not be estimated.
`param.est`	Shall be either `TRUE` or `FALSE`. `TRUE` means that `param` will be estimated.
`df`	Degrees of freedom, if not meant to be estimated. Only necessary if tested for "t"-copula.
`df.est`	Indicates if `df` shall be estimated. Has to be either `FALSE` or `TRUE`, where `TRUE` means that it will be estimated.
`m_b`	The power of the statistic.
`zeta.m`	The adjustment parameter.
`b_Rn`	The bandwidth for the estimation of the first-order partial derivatives based on the empirical copula.
`execute.times.comp`	Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if `M` is at least 100.

Details

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

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

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

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

with $C_{\theta_n}(\mathbf{u})$ an estimation of C under the H_0. $m \geq 0$ and $\zeta_m \geq 0$ are tuning parameters.

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

$$(0.5 + \sum_{b=1}^{N} 1_{\{T_b \leq T\}}/(N+1),$$

where T and T_b denote the test statistic and the bootstrapped test statistc, respectively. This ensures that the approximate p-value is a number strictly between 0 and 1, which is sometimes necessary for further treatments. See Pesarin (2001) for more details.

Value

A object of the `class` gofCOP with the components

`method` a character which informs about the performed analysis

`erg.tests` a matrix with the p-value and test statistic of test

References

Christian Genest, Wanling Huang and Jean-Marie Dufour (2013). A regularized goodness-of-fit test for copulas. *Journal de la Societe Francaise de Statistique et revue de statistique appliquee 154.1: 64-77*.

Marius Hofert, Ivan Kojadinovic, Martin Maechler, Jun Yan (2014). copula: Multivariate Dependence with Copulas. *R package version 0.999-12.*. http://CRAN.R-project.org/package=copula

Pesarin, F. (2001). *Multivariate Permutation Tests: With applications in Biostatistics*, Wiley

Examples

```
data = cbind(rnorm(100), rnorm(100))

gofRn("gaussian", data, M = 20)
```

---

gofRosenblattSnB *The SnB test based on the Rosenblatt transformation*

Description

gofRosenblattSnB is a wrapper for the functions `gofCopula`, `fitCopula`, `ellipCopula` and `archmCopula` from the package **copula**. It combines these functions to test a dataset for a copula directly without all the necessary intermediate steps and extends its functionality. The margins can be estimated by a bunch of distributions and the time which is necessary for the estimation can be given. 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 approximate p-values are obtained via parametric bootstrapping. It is possible to insert datasets of all dimensions above 1 and the possible copulae are "gaussian", "t", "gumbel", "clayton" and "frank". The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used.

Usage

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

Arguments

copula — The copula to test for. Possible are "gaussian", "t", "clayton", "gumbel" and "frank".

x — A matrix containing the residuals of the data.

M — Number of bootstrapping loops.

param — The copula parameter to use, if it shall not be estimated.

param.est — Shall be either TRUE or FALSE. TRUE means that param will be estimated.

df — Degrees of freedom, if not meant to be estimated. Only necessary if tested for "t"-copula.

df.est — Indicates if df shall be estimated. Has to be either FALSE or TRUE, where TRUE means that it will be estimated.

margins — Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp").

execute.times.comp — Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if M is at least 100.

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 $\mathbf{E}_i$, 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 $C_\perp(\mathbf{u}) = u_1 \cdot \ldots \cdot u_d$.

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

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

$$(0.5 + \sum_{b=1}^{N} 1_{\{T_b \leq T\}} / (N + 1),$$

where T and T_b denote the test statistic and the bootstrapped test statistc, respectively. This ensures that the approximate p-value is a number strictly between 0 and 1, which is sometimes necessary for further treatments. See Pesarin (2001) for more details.

Value

A object of the `class` gofCOP with the components

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

References

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

Marius Hofert, Ivan Kojadinovic, Martin Maechler, Jun Yan (2014). copula: Multivariate Dependence with Copulas. *R package version 0.999-12.*. http://CRAN.R-project.org/package=copula

Pesarin, F. (2001). *Multivariate Permutation Tests: With applications in Biostatistics*, Wiley

Examples

```
data = cbind(rnorm(100), rnorm(100), rnorm(100))

gofRosenblattSnB("gaussian", data, M = 20)
```

`gofRosenblattSnC` *The SnC test based on the Rosenblatt transformation*

Description

`gofRosenblattSnC` is a wrapper for the functions `gofCopula`, `fitCopula`, `ellipCopula` and `archmCopula` from the package **copula**. It combines these functions to test a dataset for a copula directly without all the necessary intermediate steps and extends its functionality. The margins can be estimated by a bunch of distributions and the time which is necessary for the estimation can be given. `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 approximate p-values are obtained via parametric bootstrapping. It is possible to insert datasets of all dimensions above 1 and the possible copulae are "gaussian", "t", "gumbel", "clayton" and "frank". The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used.

Usage

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

Arguments

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

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 $\mathbf{E}_i$, 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 $C_\perp(\mathbf{u}) = u_1 \cdot \ldots \cdot u_d$.

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

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

$$(0.5 + \sum_{b=1}^N 1_{\{T_b \leq T\}} / (N + 1)$$

where T and T_b denote the test statistic and the bootstrapped test statistc, respectively. This ensures that the approximate p-value is a number strictly between 0 and 1, which is sometimes necessary for further treatments. See Pesarin (2001) for more details.

Value

A object of the `class` gofCOP with the components

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

References

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

Marius Hofert, Ivan Kojadinovic, Martin Maechler, Jun Yan (2014). copula: Multivariate Dependence with Copulas. *R package version 0.999-12.*. http://CRAN.R-project.org/package=copula

Pesarin, F. (2001). *Multivariate Permutation Tests: With applications in Biostatistics*, Wiley

Examples

```
data = cbind(rnorm(100), rnorm(100), rnorm(100))

gofRosenblattSnC("gaussian", data, M = 20)
```

gofSn — *The Sn gof test using the empirical copula*

Description

gofSn is a wrapper for the functions gofCopula, fitCopula, ellipCopula and archmCopula from the package **copula**. It combines these functions to test a dataset for a copula directly without all the necessary intermediate steps and extends its functionality. The margins can be estimated by a bunch of distributions and the time which is necessary for the estimation can be given. gofSn performs the "Sn" gof test, described in Genest et al. (2009), for copulae and compares the empirical copula against a parametric estimate of the copula derived under the null hypothesis. It exist two methods to obtain the approximate p-values, parametric bootstrap and a fast multiplier approach. It is possible to insert datasets of all dimensions above 1 and the possible copulae are "gaussian", "t", "gumbel", "clayton" and "frank". The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used.

Usage

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

Arguments

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

Details

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

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

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

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

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

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

$$(0.5 + \sum_{b=1}^{N} 1_{\{T_b \leq T\}} / (N + 1),$$

where T and T_b denote the test statistic and the bootstrapped test statistc, respectively. This ensures that the approximate p-value is a number strictly between 0 and 1, which is sometimes necessary for further treatments. See Pesarin (2001) for more details.

Value

A object of the `class` gofCOP with the components

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

References

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

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

Marius Hofert, Ivan Kojadinovic, Martin Maechler, Jun Yan (2014). copula: Multivariate Dependence with Copulas. *R package version 0.999-12.*. http://CRAN.R-project.org/package=copula

Pesarin, F. (2001). *Multivariate Permutation Tests: With applications in Biostatistics*, Wiley

Examples

```
data = cbind(rnorm(100), rnorm(100), rnorm(100))

gofSn("gaussian", data, M = 20)
```

gofWhich — *Applicable gof tests for testing problem*

Description

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

Usage

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

Arguments

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

Details

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

Value

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

Examples

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

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

gofWhichCopula — *Implemented copula for a certain test*

Description

gofWhichCopula returns for a given test the applicable implemented copula.

Usage

```
gofWhichCopula(test)
```

Arguments

test	The test to search for copula.

Details

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

Value

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

Examples

```
gofWhichCopula("gofRosenblattSnB")

gofWhichCopula("gofPIOSTn")
```

---

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

Description

gofWhite is a wrapper for the function BiCopGofTest from the package **VineCopula** which extends its functionality. The margins can be estimated by a bunch of distributions and the time which is necessary for the estimation can be given. It tests a given 2 dimensional dataset for a copula with the gof test based on White's information matrix equality. The possible copulae are "gaussian", "t", "gumbel", "clayton" and "frank". See for reference Schepsmeier et al. (2015). The parameter estimation is performed with pseudo maximum likelihood method.

Usage

```
gofWhite(copula, x, M = 1000, param = 0, param.est = T, df = 0, df.est = T,
         margins = "ranks", execute.times.comp = T)
```

Arguments

copula	The copula to test for. Possible are the copulae "gaussian", "t", "clayton", "gumbel" and "frank".
x	A 2 dimensional matrix containing the residuals of the data.
M	Number of bootstrap samples.
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 for the "t"-copula.

`df.est` Indicates if `df` shall be estimated. Has to be either `FALSE` or `TRUE`, where `TRUE` means that it will be estimated.

`margins` Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is `"ranks"`, which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: `"beta"`, `"cauchy"`, Chi-squared (`"chisq"`), `"f"`, `"gamma"`, Log normal (`"lnorm"`), Normal (`"norm"`), `"t"`, `"weibull"`, Exponential (`"exp"`).

`execute.times.comp`
Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if `M` is at least 100.

Details

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

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

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

The test statistic is derived by

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

with

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

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

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

and

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

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

The test statistic will be rejected if

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

Value

A object of the `class` gofCOP with the components

`method` a character which informs about the performed analysis

`erg.tests` a matrix with the p-value and test statistic of test

References

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 = cbind(rnorm(100), rnorm(100))

gofWhite("gaussian", data)
```

Index