

gofCopula: Goodness-of-Fit Tests for Copulae

by *Ostap Okhrin, Simon Trimborn and Martin Waltz*

Abstract The last decades show an increased interest in modeling various types of data through copulae. Different copula models have been developed, which lead to the challenge of finding the best fitting model for a particular dataset. From the other side, a strand of literature developed a list of different Goodness-of-Fit (GoF) tests with different powers under different conditions. The usual practice is the selection of the best copula via the p -value of the GoF test. Although this method is not purely correct due to the fact that non-rejection does not imply acceptance, this strategy is favored by practitioners. Unfortunately, different GoF tests often provide contradicting outputs. The proposed R-package brings under one umbrella 13 most used copulae - plus their rotated variants - together with 16 GoF tests and a hybrid one. The package offers flexible margin modeling, automatized parallelization, parameter estimation, as well as a user-friendly interface, and pleasant visualizations of the results. To illustrate the functionality of the package, two exemplary applications are provided.

1 Introduction

Being firstly introduced in 1959 by Abe Sklar (see Sklar (1959)), copulae gained enormous popularity in applications in the last two decades. Researchers from different fields recognize the power of copulae while working with multivariate datasets from insurance (Fang and Madsen, 2013; Shi et al., 2016), finance (Salvatierra and Patton, 2015; Oh and Patton, 2018), biology (Konigorski et al., 2014; Dokuzoğlu and Purutçuoğlu, 2017), hydrology (Liu et al., 2018; Valle and Kaplan, 2019), medicine (Kuss et al., 2014; Gomes et al., 2019), traffic engineering, (Huang et al., 2017; Ma et al., 2017), etc. For a recent review, we refer to Größer and Okhrin (2021). Unfortunately, the correct specification of the multivariate distribution is not easy to find, and often interest in the understanding of the functional form of the copula is dominated by the expected performance of the whole model. This is natural, taking into account the huge list of different copula models proposed in the literature for different needs; see, e.g., Durante and Sempi (2010), Joe and Kurowicka (2010), or Genest and Nešlehová (2014). Although an expanding list of R-packages devoted to copulae is existent, the issue of GoF testing is less frequently addressed. Primarily, GoF tests for copulae are implemented in copula comparison packages as **copula** (Hofert et al., 2020), **TwoCop** (Remillard and Plante, 2012), and **VineCopula** (Nagler et al., 2019), but since Remillard and Scaillet (2009) and Genest et al. (2009), many other powerful tests were developed that are not integrated into these packages. Most of the tests focus on the bivariate case, leaving a further gap in the existing R-package landscape.

Given a variety of tests, the selection of the most appropriate copula seems simple at first glance. However, the power of these tests varies significantly depending on the use case and the copula tested for. The absence of the overall best GoF test leads researchers and practitioners often to the selection of the test (and copula), which supports some subjective expectation, but not the copula that fits the data at its best. Although GoF tests are not intended for model selection but rather to decide whether the selected copula is *not* suitable for the data, the model selection strategy based on the rank of the p -values is still commonly used. Following proper scoring rules (Gneiting and Raftery, 2007), some tests still allow for selection, and even if not purely statistically sound, it is heavily advocated among practitioners; see De Valpine (2014). An eloquent illustrative example of different powers and contradictory decisions was provided in Zhang et al. (2016), where three different tests (R_n by Zhang et al. (2016), S_n by Genest et al. (2009), and J_n by Scaillet (2007)) were applied for testing the dependency between the standardized returns of the Bank of America and Citigroup. The model selection was done from three copula models: normal, Gumbel, and t -copula, based on their respective p -values. Interestingly,

- 1) for the year 2004, the R_n gave a favor for the Gumbel copula, while the S_n and J_n for the normal one;
- 2) for the year 2006, the S_n gave the favor for the normal copula, while R_n and J_n for the t -copula;
- 3) for the year 2009, the J_n indicated that the dependency is close to the normal one, while R_n and S_n were in favor of the Gumbel copula.

This implies that for each year, a *different* pair of tests returns consistent results. In an empirical study, it is difficult to decide which *copula* is suitable and which *test* provides the most plausible results. An extensive comparative study of different GoF tests was performed a decade ago by Genest et al. (2009), intensively discussing all, up to that moment existing, tests for copulae. The main findings are that

there is no superior blanket test, but several tests have very good power under different, often disjunct conditions. A test proposed by Zhang et al. (2016) fills some gaps in the set of models under which this test performs better than others under certain conditions. However, a common phenomenon in empirical studies is the interpretation of the non-rejection of a copula as the correct model. Especially, in situations where the used GoF test has low power, this is not necessarily the case. Tackling this issue, Zhang et al. (2016) also developed the hybrid test, which is simple in construction and implementation. It combines the power of different tests and is very helpful for practitioners; see Section 2.6. However, even in this case, the interpretation of finding the *correct* copula should be treated with care.

We propose the R-package **gofCopula** to automatize the whole empirical procedure of selecting the most suitable copula. Table 1 displays the broad range of available tests, copula models, and the maximum dimension. The latest version of this table is also accessible via the function `CopulaTestTable()` in the package. Further details on the functionality of each test are provided in Section 2.3, while Table A.1 of Appendix A contains some characteristics of the copulae implemented in the package.

Test	normal	t	clayton	gumbel	frank	joe	amh	galambos	huslerReiss	tawn	tev	fgm	plackett
<code>gofCvM</code>	≥ 2	≥ 2	≥ 2	≥ 2	≥ 2	≥ 2	2	2	2	2	2	2	2
<code>gofKS</code>	≥ 2	≥ 2	≥ 2	≥ 2	≥ 2	≥ 2	2	2	2	2	2	2	2
<code>gofKendallCvM</code>	≥ 2	≥ 2	≥ 2	≥ 2	≥ 2	≥ 2	2	2	2	2	2	2	2
<code>gofKendallKS</code>	≥ 2	≥ 2	≥ 2	≥ 2	≥ 2	≥ 2	2	2	2	2	2	2	2
<code>gofRosenblattSnB</code>	≥ 2	≥ 2	≥ 2	≥ 2	≥ 2	≥ 2	2	2	-	-	-	2	2
<code>gofRosenblattSnC</code>	≥ 2	≥ 2	≥ 2	≥ 2	≥ 2	≥ 2	2	2	-	-	-	2	2
<code>gofRosenblattGamma</code>	≥ 2	≥ 2	≥ 2	≥ 2	≥ 2	≥ 2	2	2	-	-	-	2	2
<code>gofRosenblattChisq</code>	≥ 2	≥ 2	≥ 2	≥ 2	≥ 2	≥ 2	2	2	-	-	-	2	2
<code>gofArchmSnB</code>	-	-	≥ 2	≥ 2	≥ 2	≥ 2	2	-	-	-	-	-	-
<code>gofArchmSnC</code>	-	-	≥ 2	≥ 2	≥ 2	≥ 2	2	-	-	-	-	-	-
<code>gofArchmGamma</code>	-	-	≥ 2	≥ 2	≥ 2	≥ 2	2	-	-	-	-	-	-
<code>gofArchmChisq</code>	-	-	≥ 2	≥ 2	≥ 2	≥ 2	2	-	-	-	-	-	-
<code>gofKernel</code>	2	2	2	2	2	2	2	2	2	2	2	2	2
<code>gofWhite</code>	2	2	2	2	2	2	-	-	-	-	-	-	-
<code>gofPIOSTn</code>	3	2	3	3	3	3	2	2	-	-	-	2	2
<code>gofPIOSRn</code>	3	2	3	3	3	3	2	2	-	-	-	2	2

Table 1: Implemented tests, copula models (columns), and the maximum available dimension of each test-copula combination. "-" means this combination is not available, "2" is available in dimension two, "3" in dimensions two and three, and "≥ 2" in any dimension. `amh` corresponds to the Ali-Mikhail-Haq copula, `tev` to the *t*-extreme value copula, and `fgm` to the Farlie-Gumbel-Morgenstern copula.

In summary, the package **gofCopula** offers the following attractive features which distinguish it from other R-packages:

- Each of the 13 copulae in Table 1 is available in a rotated form for the bivariate case. Furthermore, the flexible hybrid test is implemented to aggregate the results of the 16 tests.
- We provide an interface to integrate new GoF tests. The users can provide their own test statistics and perform the tests with the integrated parametric bootstrap and also make use of the automatized parallelization of **gofCopula**. The new tests can be further combined with other tests via the hybrid test.
- The whole copula community relies justifiably on the R-package **copula** for conducting different studies on copulae. Thus we provide an interface to use objects from the R-package **copula** and perform the GoF tests with **gofCopula**.
- For the estimation of the margins, ten different parametric distributions are available, in addition to the nonparametric estimation per default.
- GoF tests rely on bootstrapping methods which can result in substantially high computational costs. In contrast to other R-packages, the package **gofCopula** comes with an integrated option for automatized parallelization of the bootstrapping samples. For the convenience of the user, the parallelization can be activated by specifying the number of parallel jobs as an argument of the functions.

- As we believe in reproducible science, the user has the opportunity to specify the seeds for the bootstrapping procedure in order to guarantee full reproducibility of all results gained from the package **gofCopula**.
- An estimation function for the computation time is implemented, which fits a regression model to give an estimate for the time adapted to the users' machine.
- An informative console output is implemented, which keeps the user informed about the current test and copula under estimation, as well as the remaining time until the derivation of the test is performed. The latter functionality is supported by the R-package **progress** (Csárdi and FitzJohn, 2019).
- The results of the tests are provided with the package's own class "gofCOP", which allows for a comprehensive overview of the test results. For a better comparison of the results, we extend the generic plot function for objects of class "gofCOP", which illustrates the results in a convenient manner. The plot function is supported by the R-package **yarr** (Phillips, 2017), and was customized to provide the user an insightful figure for the interpretation of the results.

The test statistics of six GoF tests were already implemented in R-packages. Thus, for the computation of the test statistics of some tests, we use the functions `gofTstat` and `BiCopGofTest` from the packages **copula** and **VineCopula**, respectively. For obvious reasons, we did not implement all existing tests on copulae, but we will embed new tests in the proposed package as soon as they become more relevant and actively used among academics and practitioners.

The paper, introducing the R-package **gofCopula**, is structured as follows: The tests and methodology implemented in the package are introduced in Sections 2.2 and 2.3 before presenting the functionalities of the package in Section 2.4. We explain major functions, how to apply them, and elaborate the main arguments of each function. The explanations are supported by R-code and output. To provide an impression of the runtime of various tests, we discuss the speed of the tests depending on the copula to test for, the number of observations, and the number of bootstrap samples. A simulated example (Section 2.5) contains a typical step-by-step procedure of how the package can be used in practice, which is also applied to two real-world examples (Section 2.6), in which all corresponding codes are given and explained. The cases are illustrated with interpretations of the console output and plots, both generated directly from **gofCopula**, without any additional code. The results of the two applications can be fully reproduced by the **gofCopula** package, which also contains the used datasets. All illustrations, simulations, and applications in this paper are fully reproducible and designed to guide the user into conducting their own research with the **gofCopula** package.

2 Estimation methods

Consider a d -dimensional random vector $X = \{X_1, \dots, X_d\}$ with corresponding marginal distributions $F_j, j = 1, \dots, d$. The multivariate distribution $F(x_1, \dots, x_d)$ can be decomposed via the copula function $C_\theta(u_1, \dots, u_d)$ as

$$(X_1, \dots, X_d) \sim F(x_1, \dots, x_d) = C_\theta\{F_1(x_1), \dots, F_d(x_d)\}.$$

Having a sample $\mathcal{X} = \{x_{ij}\}, i = 1, \dots, n, j = 1, \dots, d$ of size n with the columns defined as x_j , the main task for empirical studies is to estimate the copula parameter θ and the margins $F_j, j = 1, \dots, d$ for a given copula specification. Since the properties and goodness of the estimator of θ heavily depend on the estimators of the latter, their choice is also of importance.

In the bivariate case, one of the standard methods of estimation of the univariate parameter θ is based on Kendall's τ by Genest and Rivest (1993). Following Joe (1997) for (X_1, X_2) , and (X'_1, X'_2) being independent random pairs with continuous distribution F , Kendall's τ is defined via:

$$\tau = P\{(X_1 - X'_1)(X_2 - X'_2) > 0\} - P\{(X_1 - X'_1)(X_2 - X'_2) < 0\}.$$

This can be written in terms of the underlying copula C in form of $\tau = 4 \int C dC - 1$, linking Kendall's τ and the copula parameter of interest under a correct copula specification, e.g., for the normal copula, the equality $\tau = \frac{2}{\pi} \arcsin \theta$ holds, with θ being the copula correlation, c.f. Demarta and McNeil (2005). Thus, the parameter θ can be estimated via inversion of this relation and replacement of τ by its empirical counterpart. However, as shown in Genest et al. (1995), the ML method leads to substantially more efficient estimators. Therefore, we employ it as the first option in our study. The second reason for ML estimation is the fact that several implemented tests are based on the likelihood ratios. Thus, results based on Kendall's τ will not be supported by the theory behind these tests. The ML procedure

can be performed for the parameters of the margins and of the copula function simultaneously:

$$(\hat{\theta}, \hat{\alpha}_1, \dots, \hat{\alpha}_d)^\top = \operatorname{argmax}_{\theta, \alpha_1, \dots, \alpha_d} \mathcal{L}(\mathcal{X}, \theta, \alpha_1, \dots, \alpha_d), \tag{1}$$

$$\text{with } \mathcal{L}(\mathcal{X}, \theta, \alpha_1, \dots, \alpha_d) = \sum_{i=1}^n \log \left[c\{F_1(x_{i1}, \alpha_1), \dots, F_d(x_{id}, \alpha_d), \theta\} \prod_{j=1}^d f_j(x_{ij}, \alpha_j) \right], \tag{2}$$

where $c(\cdot)$ is the copula density, $\alpha_1, \dots, \alpha_d$ are parameters of the margins, $f_j(\cdot)$ are marginal densities, and $\mathcal{L}(\cdot)$ is the full log-likelihood function. Nevertheless, simultaneous maximization of the function in (1) is very computationally intensive. Therefore, we consider only two-stage procedures, where at the first stage, we estimate margins parametrically (c.f. Joe (1997) and Joe (2005)) as

$$F_j(x, \hat{\alpha}_j) = F_j \left\{ x, \operatorname{argmax}_{\alpha} \sum_{i=1}^n \log f_j(x_{ij}, \alpha) \right\}, \quad \text{for } j = 1, \dots, d, \tag{3}$$

or nonparametrically (c.f. Chen and Fan (2006) and Chen et al. (2006)) as

$$\hat{F}_j(x) = (n + 1)^{-1} \sum_{i=1}^n \mathbf{1}\{x_{ij} \leq x\}, \quad j = 1, \dots, d, \tag{4}$$

with $\mathbf{1}$ being the indicator function. Afterward, the copula parameter is estimated in the second step as

$$\hat{\theta} = \operatorname{argmax}_{\theta} \sum_{i=1}^n \log c\{\tilde{F}_1(x_{i1}), \dots, \tilde{F}_d(x_{id}), \theta\}, \tag{5}$$

where $\tilde{F}(x) \in \{\hat{F}(x), F(x, \hat{\alpha})\}$ are parametrically or nonparametrically estimated margins. In the case of parametric margins, one shall be aware that the two-step approach does not lead to efficient estimators, though the loss in the efficiency is moderate and mainly depends on the strength of dependencies (Joe, 1997). The method of nonparametric estimation of the marginal distributions for copula estimation was first used in Oakes (1994) and further investigated in Genest et al. (1995) and Shih and Louis (1995).

Furthermore, Fermanian and Scaillet (2003) and Chen and Huang (2007) consider a fully non-parametric estimation of the copula, which is heavily used in the GoF testing. It is called an empirical copula and is shown to be a consistent estimator of the true underlying copula, c.f. Gaensler and Stute (1987) and Radulovic and Wegkamp (2004). This estimator is defined as

$$C_n(u_1, \dots, u_d) = n^{-1} \sum_{i=1}^n \prod_{j=1}^d \mathbf{1}\{\hat{F}_j(x_{ij}) \leq u_j\}.$$

3 Goodness-of-fit tests for copulae

Having a list of different copulae, the most suitable one for real applications still needs to be found and motivated. For this purpose, a series of different GoF tests has been developed in the last decades. Several authors, e.g., Genest et al. (2009), tested the power of those tests against each other and showed that no superior test for all possible situations exist. We cover 16 tests and implement them into the **gofCopula** package. Most of these tests work with the parametric family of copulae denoted by $C_0 = \{C_\theta; \theta \in \mathbb{A} \subset \mathbb{R}^p\}$ for some integer $p \geq 1$ and the copula C , under the general H_0 -hypothesis:

$$H_0 : C \in C_0.$$

We differentiate seven groups of GoF tests for copulae based on: (1) empirical copula process; (2) Kendall’s process; (3) Rosenblatt integral transform; (4) transformation for Archimedean copulae; (5) Kernel density; (6) White’s information matrix equality; and (7) pseudo in-and-out-of-sample (PIOS) estimator.

Empirical copula process

The first group is based on the most natural approach: the deviation of the empirical copula C_n from the parametric copula $C(u_1, \dots, u_d; \theta)$, captured by the empirical copula process $\sqrt{n}\{C_n(u_1, \dots, u_d) - C(u_1, \dots, u_d; \theta)\}$. Based on an estimation of the parametric copula $C(u_1, \dots, u_d; \hat{\theta})$, the following

process can be defined:

$$C_n(u_1, \dots, u_d) = \sqrt{n}\{C_n(u_1, \dots, u_d) - C(u_1, \dots, u_d; \hat{\theta})\}.$$

Different measures of divergence can be constructed to evaluate C_n ; see Fermanian (2005) and Genest and Rémillard (2008). We implemented two commonly applied approaches using the Cramér-von Mises and Kolmogorov-Smirnov statistics:

$$S_n^E = \int_{[0,1]^d} C_n(u_1, \dots, u_d)^2 dC_n(u_1, \dots, u_d), \quad T_n^E = \sup_{u_1, \dots, u_d \in [0,1]^d} |C_n(u_1, \dots, u_d)|.$$

Notice that the Cramér-von Mises statistic yields better performances in most cases (Genest et al., 2009). The evaluation of the d -dimensional integral in practice uses numerical approximations, and the test statistic S_n^E has been already implemented in the `copula` package as function `gofTstat`, so we included it into our package. The tests are later denoted by `gofCvM` and `gofKS`, respectively.

Kendall's process

The tests from the second group were developed and investigated by Genest and Rivest (1993), Wang and Wells (2000), Genest et al. (2006). The main idea behind them is to use the copula-based random variable:

$$C\{F_1(X_1), \dots, F_d(X_d); \theta\} \sim K(\cdot, \theta), \tag{6}$$

where $K(\cdot, \theta)$ is the univariate Kendall's distribution (not uniform in general); see Barbe et al. (1996), Jouini and Clemen (1996). The empirical version of $K(\cdot)$ is given through:

$$K_n(v) = n^{-1} \sum_{i=1}^n \mathbf{1} [C_n\{\hat{F}_1(x_{i1}), \dots, \hat{F}_d(x_{id})\} \leq v], \quad v \in [0, 1].$$

Based on the definition of Kendall's process $\sqrt{n}\{K_n(v) - K(\cdot, \hat{\theta})\}$ and a parametric $K(\cdot, \hat{\theta})$ estimated with the parameter $\hat{\theta}$, we can define an empirical process as

$$\mathbb{K}_n(v) = \sqrt{n}\{K_n(v) - K(v, \hat{\theta})\}. \tag{7}$$

On this basis, two applicable test statistics are Cramér-von Mises and Kolmogorov-Smirnov; see Genest et al. (2006).

$$S_n^{(K)} = \int_0^1 \mathbb{K}_n(v)^2 dK(v, \hat{\theta}), \quad T_n^{(K)} = \sup_{v \in [0,1]} |\mathbb{K}_n(v)|.$$

Worth mentioning are the different null hypotheses $H_0'' : K \in \mathcal{K}_0 = \{K(\cdot, \theta) : \theta \in \Theta\}$ of these tests. Since $H_0 \subset H_0''$, the non-rejection of H_0'' does not imply non-rejection of H_0 . However, for bivariate Archimedean copulae, H_0'' and H_0 are equivalent (Genest et al., 2009). Both tests are later denoted as `gofKendallCvM` and `gofKendallKS`, respectively.

Rosenblatt transform

Under the assumption of copula dependency, the conditional distribution of U_i given U_1, \dots, U_{i-1} is specified through:

$$\begin{aligned} C_d(u_i|u_1, \dots, u_{i-1}) &= P\{U_i \leq u_i | U_1 = u_1 \dots U_{i-1} = u_{i-1}\} \\ &= \frac{\partial^{i-1} C(u_1, \dots, u_i, 1, \dots, 1) / \partial u_1 \dots \partial u_{i-1}}{\partial^{i-1} C(u_1, \dots, u_{i-1}, 1, \dots, 1) / \partial u_1 \dots \partial u_{i-1}}. \end{aligned}$$

The Rosenblatt transform (c.f. Rosenblatt, 1952) is then defined as follows.

Definition 1 Rosenblatt's probability integral transform of a copula C is the mapping $\mathfrak{R} : (0, 1)^d \rightarrow (0, 1)^d$, $\mathfrak{R}(u_1, \dots, u_d) = (e_1, \dots, e_d)$ with $e_1 = u_1$ and $e_i = C_d(u_i|u_1, \dots, u_{i-1})$, $\forall i = 2, \dots, d$.

If the copula is correctly specified, the variables $(e_1, \dots, e_d)^\top$ resulting from the Rosenblatt transform should be independent from each other and uniformly distributed. Therefore, the null hypothesis $H_0 : C \in \mathcal{C}_0$ is equivalent to

$$H_{0R} : (e_1, \dots, e_d)^\top \sim \Pi, \tag{8}$$

where $\Pi(u_1, \dots, u_d) = u_1 \dots u_d$ is the product (independence) copula.

Two different types of tests may be constructed using this property. In the first type, similar to the previous two groups, we measure the deviation of the product copula of $(e_1, \dots, e_d)^\top$ from the corresponding empirical copula:

$$D_n(u_1, \dots, u_d) = n^{-1} \sum_{i=1}^n \prod_{j=1}^d \mathbf{1}\{e_{ij} \leq u_j\}.$$

Thus, following Genest et al. (2009), two Cramér-von Mises statistics result:

$$S_n^{(B)} = n \int_{[0,1]^d} \{D_n(u_1, \dots, u_d) - \Pi(u_1, \dots, u_d)\}^2 du_1 \cdots du_d,$$

$$S_n^{(C)} = n \int_{[0,1]^d} \{D_n(u_1, \dots, u_d) - \Pi(u_1, \dots, u_d)\}^2 dD_n(u_1, \dots, u_d).$$

Since the H_0 changed to H_{0R} , the tests evaluate the difference of $D_n(u)$ to the product copula. In the package, these tests are defined as `gofRosenblattSnB` and `gofRosenblattSnC`, respectively.

The second type of test uses the fact that a specific combination of independent uniformly distributed random variables follows some known distribution. Based on this, two further Anderson-Darling type tests were introduced by Breymann et al. (2003). By defining

$$G_{i,\Gamma} = \Gamma_d \left\{ \sum_{j=1}^d (-\log e_{ij}) \right\},$$

where $\Gamma_d(\cdot)$ is the Gamma distribution with shape d and scale 1 and

$$G_{i,\chi^2} = \chi_d^2 \left[\sum_{j=1}^d \{\Phi^{-1}(e_{ij})\}^2 \right],$$

where $\chi_d^2(\cdot)$ is the Chi-squared distribution with d degrees of freedom and Φ being the standard normal distribution. It results:

$$T_n = -n - \sum_{i=1}^n \frac{2i-1}{n} [\log G_{(i)} + \log\{1 - G_{(n+1-i)}\}],$$

where $G_{(i)}$ is the i -th ordered observation of the $G_{i,\Gamma}$ or G_{i,χ^2} . One should note that Anderson-Darling type tests have almost no power and even do not capture the type 1 error (Dobrić and Schmid, 2007), while the Cramér-von Mises tests behave much more satisfactory (Genest et al., 2009). Furthermore, the basic assumption of uniformly distributed and independent observations after applying the Rosenblatt transform is violated since those variables are not mutually independent and only approximately uniform. The latter two tests are denoted in the package as `gofRosenblattGamma` and `gofRosenblattChisq`, respectively, and are obtained via the function `gofTstat` from the package `copula`.

Transformation for Archimedean copulae

Recently, Hering and Hofert (2015) proposed a procedure of GoF testing based on a transformation similar to the one of Rosenblatt (1952) specifically designed for Archimedean copulae.

Definition 2 *Hering and Hofert's transformation of an Archimedean copula C of dimension $d \geq 2$ with d -monotone generator ψ and continuous Kendall distribution K is the mapping $\mathfrak{T} : (0, 1)^d \rightarrow (0, 1)^d$,*

$$\mathfrak{T}(u_1, \dots, u_d) = (v_1, \dots, v_d) \text{ with } v_i = \left\{ \frac{\sum_{k=1}^i \psi^{-1}(u_k)}{\sum_{k=1}^{i+1} \psi^{-1}(u_k)} \right\}^i, \forall i = 1, \dots, d-1 \text{ and } v_d = K\{C(u_1, \dots, u_d)\}.$$

Distribution function K is estimated empirically, and the variables $(v_1, \dots, v_d)^\top$ are independent and uniformly distributed if the copula is correctly specified. Note that this transformation was originally considered in Wu et al. (2007) as a method for generating random numbers from Archimedean copulae, such as the inverse of the Rosenblatt transform can be used for sampling copulae. Following Hering and Hofert (2015), the main advantage of this approach in comparison to tests based on the Rosenblatt transform is the more convenient computation in higher dimensions, in which the Rosenblatt procedure is numerically challenging and unstable.

The null hypothesis equals (8) from the tests based on Rosenblatt's probability integral transform: $H_{0T} : (v_1, \dots, v_d)^\top \sim \Pi$ with Π being the product copula. Consequently, the approaches to test it are

identical. In analogy to the naming introduced in Section 2.3.3, we denoted the tests as `gofArchmSnB`, `gofArchmSnC`, `gofArchmGamma`, and `gofArchmChisq` in the package.

Kernel density estimator

A test from this group has been introduced by [Scaillet \(2007\)](#). Following his approach, a d -variate quadratic kernel \mathcal{K} with bandwidth $H = 2.6073n^{-1/6}\widehat{\Sigma}^{1/2}$ is used, with $\widehat{\Sigma}$ being a sample covariance matrix with $\widehat{\Sigma}^{1/2}$ its Cholesky decomposition. Using $\mathcal{K}_H(y_1, \dots, y_d) = \mathcal{K}(H^{-1}\{y_1, \dots, y_d\}^\top) / \det(H)$, the copula density is nonparametrically estimated by

$$\hat{c}(u_1, \dots, u_d) = n^{-1} \sum_{i=1}^n \mathcal{K}_H[(u_1, \dots, u_d)^\top - \{\tilde{F}_1(x_{i1}), \dots, \tilde{F}_d(x_{id})\}^\top],$$

where under $\tilde{F}_i(\cdot)$, we consider nonparametric as well as parametric estimators of the margins. The test statistic is then:

$$J_n = \int_{[0,1]^d} \{\hat{c}(u_1, \dots, u_d) - \mathcal{K}_H * c(u_1, \dots, u_d; \hat{\theta})\}^2 w(u_1, \dots, u_d) du_1 \cdots du_d, \tag{9}$$

with “ $*$ ” being a convolution operator, $w(u_1, \dots, u_d)$ a weight function, and $c(u_1, \dots, u_d; \hat{\theta})$ the copula density under the H_0 , with estimated copula parameter $\hat{\theta}$. Note that the integral is computed numerically using the Gauss-Legendre quadrature method; see [Scaillet \(2007\)](#). The number of knots can be specified via the argument `nodes`. Integration. A scaling parameter for H is implemented via `delta`. `J`, and the internal size of the bootstrapping samples can be controlled via `MJ`. This test is denoted by `gofKernel` in the package.

White test

This test was introduced by [Huang and Prokhorov \(2014\)](#) and had its foundation in the information matrix equality stated by [White \(1982\)](#). Given the presence of certain regularity conditions, the White equality establishes a connection between the negative sensitivity matrix $S(\theta)$, and the variability matrix $V(\theta)$ defined as

$$S(\theta) = -E_0 \left[\frac{\partial^2}{\partial \theta \partial \theta^\top} \log c\{F_1(x_1), \dots, F_d(x_d); \theta\} \right],$$

$$V(\theta) = E_0 \left(\left[\frac{\partial}{\partial \theta} \log c\{F_1(x_1), \dots, F_d(x_d); \theta\} \right] \left[\frac{\partial}{\partial \theta} \log c\{F_1(x_1), \dots, F_d(x_d); \theta\} \right]^\top \right),$$

where E_0 is the expectation under correct model specification, which is represented by the null hypothesis to be specified. The equality states:

$$S(\theta) = V(\theta).$$

Using this approach, the testing problem can be formulated as follows:

$$H_{0W} : S(\theta) = V(\theta) \quad \text{vs.} \quad H_{1W} : S(\theta) \neq V(\theta).$$

Following [Schepsmeier \(2015\)](#), a test statistic is based on empirical versions of the two information matrices, denoted by $\widehat{S}(\hat{\theta})$ and $\widehat{V}(\hat{\theta})$. These are aggregated via $d(\hat{\theta}) = \text{vech}\{\widehat{S}(\hat{\theta}) + \widehat{V}(\hat{\theta})\}$ with `vech` denoting vectorization of the lower triangular of a matrix. As a result, $d(\hat{\theta})$ is a vector of dimension $\frac{p(p+1)}{2}$, given the copula parameter vector is of dimension p . It can be shown that the constructed test statistics:

$$T_W = n\{d(\hat{\theta})\}^\top \hat{A}_\theta^{-1} d(\hat{\theta}),$$

with \hat{A}_θ^{-1} being the sample estimator of the asymptotic covariance matrix of $\sqrt{nd}(\hat{\theta})$, follows asymptotically a χ^2 distribution with $\frac{p(p+1)}{2}$ degrees of freedom. For the derivation of the test statistic, this test relies on the function `BiCopGofTest` from the **VineCopula** package, again, in order to avoid code redundancy. Note that the implementation of the test can be unstable for the t -copula; see [Nagler et al. \(2019\)](#). This is the reason why it could not be computed in some cases of the second empirical example in Section 2.6.2. This test is called `gofWhite` in the package.

Cross-validated tests

A recent test using a leave-one-block strategy, and its approximation were introduced by Zhang et al. (2016). Authors derive $\hat{\theta}$ as in (5) and compare it with $\hat{\theta}_{-b}$, $1 \leq b \leq B$, which are delete-one-block pseudo ML estimates:

$$\hat{\theta}_{-b} = \operatorname{argmax}_{\theta \in \Theta} \sum_{b' \neq b}^B \sum_{i=1}^m \log c\{\tilde{F}_1(x_{i1}), \dots, \tilde{F}_d(x_{id}); \theta\}, \quad b = 1, \dots, B,$$

where B is the number of non-overlapping blocks and m the length of each block. Note that in the general setting, these blocks need not be of the same size. However, we follow here the approach of Zhang et al. (2016), who restrict themselves to the same length case of each block. This assumption also simplifies the usage in terms of many parameters. The resulting test statistics,

$$T_n(m) = \sum_{b=1}^B \sum_{i=1}^m \left[\log \frac{c\{\tilde{F}_1(x_{i1}), \dots, \tilde{F}_d(x_{id}); \hat{\theta}\}}{c\{\tilde{F}_1(x_{i1}), \dots, \tilde{F}_d(x_{id}); \hat{\theta}_{-b}\}} \right], \tag{10}$$

compares the full likelihood, “in-sample”, against the resulting likelihoods from the leave-one-block out estimation, “out-of-sample”. If the data in each block significantly influence the estimation of the copula parameter under the null hypothesis, then the chosen copula model is inadequate to represent the data.

Depending on the number of blocks, B , a possibly huge amount of dependence parameter estimations have to be performed to get (10). In the case of equal length of each block, $\lfloor \frac{n}{m} \rfloor$ parameters should be computed. To overcome this drawback, under suitable regularity conditions, Zhang et al. (2016) proposed the test statistic asymptotically equivalent to (10):

$$R_n = \operatorname{tr}\{\hat{S}(\hat{\theta})^{-1} \hat{V}(\hat{\theta})\}. \tag{11}$$

As we see, this result is very similar to the White (1982) test, but the power of the test is much higher. Both exact and asymptotic test statistics are denoted in the package as `gofPIOSTn` and `gofPIOSRn`, respectively.

Hybrid test

Many power studies including Genest et al. (2009) showed that no overall single optimal test exists for testing for copula models. Zhang et al. (2016) introduced a Hybrid test to combine the testing power of several tests. Having q different tests and the corresponding p -values, $p^{(1)}, \dots, p^{(q)}$, the combined p -value is defined to be:

$$p^{hybrid} = \min\{q \cdot \min(p^{(1)}, \dots, p^{(q)}), 1\}. \tag{12}$$

In Zhang et al. (2016), it is shown that the consistency of (12) is ensured as long as at least one of the q tests is consistent.

Bootstrapping test statistics

As the distribution of the test statistics is in most cases unknown, we perform a parametric bootstrap to receive the p -values. The necessary steps are described as follows:

- Step 1. Generate bootstrap sample $\{\epsilon_i^{(m)}, i = 1, \dots, n\}$ from copula $C(u_1, \dots, u_d; \hat{\theta})$ under H_0 with $\hat{\theta}$ and estimated marginal distributions \tilde{F} obtained from original data;
- Step 2. Based on $\{\epsilon_i^{(m)}, i = 1, \dots, n\}$ from Step 1, estimate θ of the copula under H_0 and compute test statistics under consideration, say ;
- Step 3. Repeat M -times Steps (1. – 2.) and obtain M statistics $T_n^m, m = 1, \dots, M$;
- Step 4. Compute an empirical p -value as $p_e = M^{-1} \sum_{m=1}^M \mathbf{1}\{|T_n^m| \geq |T_n|\}$ with T_n being the test statistics estimated from original dataset.

Depending on the different tests, variants of the described steps have to be performed. For example, the Kernel density estimation test of Scaillet (2007) described in Section 2.3.5 relies on a double bootstrapping procedure, in which for the computation of each test statistic, T_n and T_n^m in the steps above, an additional bootstrapping is utilized. Thus, the double bootstrapping approach consists of one bootstrap to calculate the p -value from a given test statistic and a second bootstrap to calculate the test statistic from an estimated copula. For further details, we refer to Scaillet (2007). Both bootstrapping procedures can be controlled via the arguments `M` and `MJ`, respectively.

4 Functionality of the R-package

The core of the **gofCopula** package is the function `gof`, which computes different tests for different copulae for a given dataset, based on the user's choice.

```
R> library("gofCopula")
R> data("IndexReturns2D", package = "gofCopula")
R> system.time(result <- gof(IndexReturns2D, M = 100, seed.active = 1))
```

The margins will be estimated as: ranks

normal copula

```
Test gofCvM is running
Test gofKendallCvM is running
Test gofKendallKS is running
Test gofKernel is running
Test gofKS is running
```

t copula

```
Test gofCvM is running
Test gofKendallCvM is running
Test gofKendallKS is running
Test gofKernel is running
Test gofKS is running
```

clayton copula

```
Test gofCvM is running
Test gofKendallCvM is running
Test gofKendallKS is running
Test gofKernel is running
Test gofKS is running
```

gumbel copula

```
Test gofCvM is running
Test gofKendallCvM is running
Test gofKendallKS is running
Test gofKernel is running
Test gofKS is running
```

frank copula

```
Test gofCvM is running
Test gofKendallCvM is running
Test gofKendallKS is running
Test gofKernel is running
Test gofKS is running
```

joe copula

```
Test gofCvM is running
Test gofKendallCvM is running
Test gofKendallKS is running
Test gofKernel is running
Test gofKS is running
```

amh copula

The copula amh is excluded from the analysis since the parameters do not fit its parameter space. See warnings and manual for more details.

galambos copula

```
Test gofCvM is running
Progress: [===>-----] 15% | time left: 3s
...
```

```
   user system elapsed
629.26   1.08  628.94
```

Warnings:

...

gof considers all 13 available copula models if no copulae or tests are specified. If a copula is unsuitable in the sense that the estimated parameter is at the boundary of the parameter space, the copula is automatically excluded, and the user is informed via a console statement (see above) and additional warnings. In the given example, this is the case for the AMH, tawn, and FGM copulae because the used IndexReturns2D dataset exhibits an estimated Kendall's $\hat{\tau} = 0.611$, which none of these three copulae can model adequately; see Table A.1. The object result is of class "gofCOP" and has length 10, which is the number of copulae used in testing (here: 13) minus the ones excluded during calculation (here: 3). Following Table 1, five tests are available for all of these copula models in $d = 2$, and these are used in the given function call.

If the user specifies copulae, tests, or both, the intersection of possible tests and copulae following Table 1 is considered. For example, if `copula = c("normal", "tawn")` is specified, the function calculates the five tests which are implemented for both copulae (assuming $d = 2$). If, on the other hand, `tests = c("gofKernel", "gofArchmSnB")` is selected, the five Archimedean copulae implemented for both tests are computed. In the case when both copulae and tests are defined, the function provides results for the possible combinations. During the calculation, the user is informed about the computation progress by statements about the running test and copula. Furthermore, a progress bar indicates the percentage of progress for this specific test as well as a dynamically updated estimated remaining time.

```
R> result$normal
```

```
$method
```

```
[1] "Parametric bootstrap goodness-of-fit test with hybrid test and normal copula"
```

```
$copula
```

```
[1] "normal"
```

```
$margins
```

```
[1] "ranks"
```

```
$param.margins
```

```
list()
```

```
$theta
```

```
[,1]
```

```
[1,] 0.8347428
```

```
$df
```

```
NULL
```

```
$res.tests
```

	p.value	test statistic
CvM	1.00	0.01520542
KendallCvM	0.41	0.06286712
KendallKS	0.11	0.80800000
Kernel	0.39	0.56012429
KS	1.00	0.31392428
hybrid(1, 2)	0.82	NA
hybrid(1, 3)	0.22	NA

...

The first element of `result` provides results for the normal copula. Note that in the field `res.tests` the hybrid tests, starting after the individual ones, contain numbers in brackets indicating which tests are considered for this hybrid. Thus, `hybrid(1, 2)` means that this is the hybrid of CvM and KendallCvM tests. The p -value 0.82 in testing for normality is obtained following formula (12) and therefore is $\min\{2 \cdot \min(1.00, 0.41), 1\} = 0.82$. To access the rotated versions of the copulae, one can set, for example, `copula = c("clayton", "gumbel")` together with `flip = c(0, 180)`, which would test for the Clayton copula and the 180 degrees rotated Gumbel copula.

gofCOP class

Objects of class "gofCOP" are generated by the function `gof` or a single test function like, e.g., `gofPIOSTn`. They consist of different sub-elements - one for each copula - as, e.g., `result$normal` in the given example. These sub-elements are lists of length seven and contain the estimation and test results for the specific copula. They present in the field `method` a description of the test scenario. The field `margins` lists the defined marginal distribution that can also be a vector of distributions where each element is applied to the respective data column, whereas `param.margins` returns the estimates of the parameters of the marginal distributions if a parametric approach was specified. Field `theta` contains the ML estimate of the copula parameter. In case of the t - and t -EV-copulae, the section `df` is the estimated number of degrees of freedom for the copula. The values of these parameters are identical for all the tests. In `res.tests`, the p -values and test statistics (only for individual tests) are given for each of the executed tests. Each row corresponds to one test from the individual to the hybrid tests. p -values of all the individual tests are computed via the bootstrap method described in Section 2.3.9. The number of bootstrap samples M can be adjusted via the parameter `M`.

Plotting gofCOP class objects

"gofCOP" objects can be called by a generic plot function allowing the user to get the p -values of the single, and the hybrid tests visualized in a pirateplot of the R-package `yarr`. It enables the user to select which copulae and hybrid testing sizes are desired for plotting. The remaining customization options are equal to those of the function `pirateplot` from the package `yarr`, except for the arguments `formula`, `data`, `sortx`, `xaxt`, `xlim`, `ylim`, and `ylob`.

```
R> plot(result, copula = c("clayton", "joe", "plackett"), hybrid = c(1, 3, 5))
```

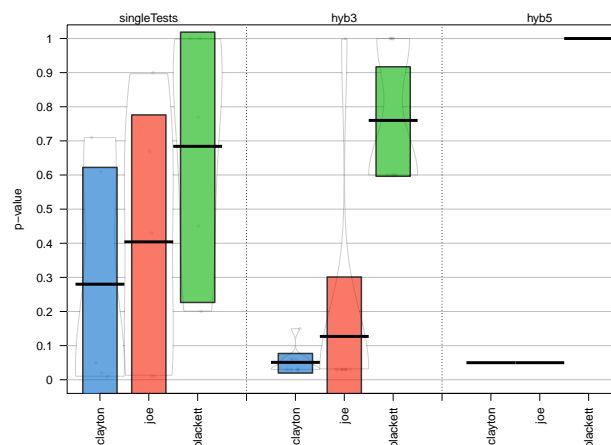


Figure 1: Resulting p -values of different hybrid tests for the Clayton, Joe, and Plackett copula visualized in a pirateplot.

Specifying `hybrid = c(1, 3, 5)` means that the p -values of the single tests (column `singleTests` in Figure 1), the p -values of hybrid tests of size three (column `hyb3`), and size five (column `hyb5`) should be plotted, separated by selected copulae. For example, we focus on the column `hyb3` for the Plackett copula. It contains information of all hybrid tests, which include three single tests for the Plackett copula. In this case, we can see that the mean of these tests is approximately 0.76, as shown by the thick horizontal line. All test p -values are shown by light-grey points in the column, indicating the heterogeneity of the tests ranging from 0.6 to 1. Finally, the green bar around the mean line is a Bayesian highest density interval, which provides the user, together with the shown density estimate in the grey continuous lines, further information about the distribution of the p -values. For more details on the `pirateplot` and its customization options, we refer to Phillips (2017).

Fire-and-forget

The R-package `gofCopula` includes all the discussed tests in Section 2.3. For each of the tests, a separate function is implemented with a variety of arguments. We give shortly the most important arguments all the tests share before we go into details about the structure of the package.

- `copula`: The copula to test for. Possible options depend on the test and dimension.

- `x`: A matrix containing the data with rows being observations and columns being variables.
- `M` (default: 1000): Number of bootstrapping loops.
- `param` (default: 0.5): The copula parameter to use if it shall not be estimated. In case of the Gumbel copula, the default value is set to 1.5.
- `param.est` (default: TRUE): Boolean. TRUE means that `param` will be estimated.
- `margins` (default: ranks): Specifies which estimation method shall be used. The default ranks stands for formula (4), which is the standard approach to convert data. Alternatively, the following distributions can be specified: beta, cauchy, Chi-squared (`chisq`), `f`, gamma, Log normal (`lnorm`), Normal (`norm`), `t`, `weibull`, Exponential (`exp`).
- `flip` (default: 0): The parameter to rotate the copula by 90, 180, 270 degrees clockwise. Only applicable for bivariate copula.
- `seed.active` (default: NULL): Sets the seeds for the bootstrapping procedure. It has to be either an integer or a vector of `M+1` integers. If an integer is provided, the seeds for the bootstrap samples will be simulated based on it. If `M+1` seeds are given, these are used in the bootstrapping procedure. In the default case (`seed.active = NULL`), R generates the seeds from the computer runtime.
- `processes` (default: 1): The number of parallel processes which are performed to speed up the bootstrapping. Should not be larger than the number of logical processors.

The package is coded as a *fire-and-forget* package. Each of the single tests just requires the input of a dataset `x` and a copula to test for. All the other function parameters have reasonable default values such that quick first results can be achieved easily. The calculation steps of each GoF test function are the following:

- *Estimation of margins*: At first, the function transforms the data nonparametrically or parametrically to $U[0, 1]$; see Section 2.2. This transformation is performed automatically, and a console statement informs the user about the transformation.
- *Estimation of copula*: Afterward, the parameters of the copula model are estimated, so a two-stage estimation is applied; see Section 2.2. Since a full ML estimation is computationally demanding, canonical ML estimation or inference for margins is applied. In case the ML estimation fails, the package automatically changes to inversion of Kendall's tau (see Section 2.2), which guarantees a result. The user is informed about that switch by a warning message.
- *Bootstrapping*: Following the estimation of the copula parameters, the bootstrapping procedure will be performed, and the empirical p -value will be derived according to the test statistics in Section 2.3.9. Since the bootstrapping procedure can require a long computational time, it can pay out to parallelize the bootstrapping via the argument `processes`.

Hybrid testing and further functionality

Besides `gof` and the single tests, the package **gofCopula** offers additional functionality for the user. Next to descriptions, illustrative examples are provided, assuming the following was called beforehand:

```
R> library("gofCopula")
R> data("IndexReturns2D", package = "gofCopula")
R> (res <- gof(copula = "normal", x = IndexReturns2D, M = 10, seed.active = 1,
+           tests = c("gofPIOSRn", "gofCvM", "gofKernel")))
```

```
The margins will be estimated as: ranks
normal copula
Test gofPIOSRn is running
Test gofCvM is running
Test gofKernel is running
```

```
-----
Parametric bootstrap goodness-of-fit test with hybrid test and normal copula
```

```
Parameters:
theta.1 = 0.834742824340301
```

```
Tests results:
           p.value test statistic
PIOSRn      0.5      -0.11032857
```

Sn	1.0	0.01520542
Kernel	0.3	0.56012429
hybrid(1, 2)	1.0	NA
hybrid(1, 3)	0.6	NA
hybrid(2, 3)	0.6	NA
hybrid(1, 2, 3)	0.9	NA

Please use the functions `gofGetHybrid()` and `gofOutputHybrid()` for display of subsets of Hybrid tests. To access the results, please obtain them from the structure of the `gofCOP` object.

The functions are:

- `gofCustomTest`: This function has - next to the standard arguments listed in Section 2.4.3 - the argument `customTest`, a character string referencing one customized test loaded in the workspace. The function containing this test should have two arguments: a matrix named `x` for the dataset and a character string named `copula` for the copula to test for. The whole calculation process, including the estimation of the margins and the copula, the calculation of the test statistics, and the bootstrapping of the p -value, is performed for this customized test. The procedure is shown using the test statistics of the `gofCvM` test.

```
R> 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)
+ }
```

```
R> gofCustomTest(copula = "normal", x = IndexReturns2D, M = 10,
+               customTest = "Testfunc", seed.active = 1)
```

The margins will be estimated as: ranks

Parametric bootstrap goodness-of-fit test with Testfunc test and normal copula

Parameters:

theta.1 = 0.834742824340301

Tests results:

	p.value	test statistic
Testfunc	1	0.01520542

- `gofGetHybrid`: Allows calculating hybrid test p -values for given p -values from customized tests with an object of class "gofCOP" generated in the package. Through the combination of `gofCustomTest` and `gofGetHybrid`, the users are not limited to the implemented tests in the package and have the opportunity to include their own tests in the analysis. Note that the function `gofOutputHybrid` has slightly different but comparable functionality, which is the reason it is not separately shown.

```
R> gofGetHybrid(result = res, nsets = 5, p_values = c("MyTest" = 0.7,
+           "AnotherTest" = 0.3))
```

Hybrid test p -values for given single tests.

Parameters:

theta.1 = 0.834742824340301

Tests results:

	p.value
PIOSRn	0.5
Sn	1.0
Kernel	0.3
MyTest	0.7
AnotherTest	0.3
hybrid(1, 2, 3, 4, 5)	1.0

- `gofTest4Copula`: Returns for a given copula and a given dimension the list of applicable implemented tests.

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

[1] "gofArchmChisq"      "gofArchmGamma"     "gofArchmSnB"
[4] "gofArchmSnC"       "gofCustomTest"     "gofCvM"
[7] "gofKendallCvM"     "gofKendallKS"      "gofKS"
[10] "gofRosenblattChisq" "gofRosenblattGamma" "gofRosenblattSnB"
[13] "gofRosenblattSnC"
```

- `gofCopula4Test`: Returns for a given test the list of applicable implemented copulae.

```
R> gofCopula4Test("gofPIOSTn")

[1] "normal"  "t"       "clayton" "gumbel"  "frank"   "joe"
[7] "amh"     "galambos" "fgm"     "plackett"
```

- `gofCheckTime`: Estimates the time necessary to compute a selected single or group of GoF tests for a given number of bootstrapping rounds. This function uses an underlying regression model, so the results may vary from reality and also from the progress bar predictions. See Section 2.4.5.

```
R> gofCheckTime("normal", x = IndexReturns2D, tests = "gofRosenblattSnC",
+               M = 10000, seed.active = 1)
```

The margins will be estimated as: ranks
 An estimate of the computational time is under derivation.
 Depending on the tests chosen, dimensionality and complexity of the data, this might take a while.
 The computation will take approximately 0 d, 0 h, 10 min and 7 sec.

- `gofco`: In the case a copula is already estimated with the package **copula**, one can provide an object of class "copula" to this function, and the parameter estimates are taken from the respective object.

```
R> copObject = normalCopula(param = 0.8)
R> gofco(copObject, x = IndexReturns2D, M = 10, seed.active = 1,
+       tests = c("gofPIOSRn", "gofKernel"))
```

The margins will be estimated as: ranks
 Test gofPIOSRn is running
 Test gofKernel is running

 Parametric bootstrap goodness-of-fit test with hybrid test and normal copula

Parameters:
 theta.1 = 0.8

Tests results:

	p.value	test statistic
PIOSRn	0.9	-0.03641543
Kernel	0.2	0.57115224
hybrid(1, 2)	0.4	NA

Please use the functions `gofGetHybrid()` and `gofOutputHybrid()` for display of subsets of Hybrid tests. To access the results, please obtain them from the structure of the `gofCOP` object.

Computational time and parallelization

One of the main drivers of long computation times is the high number of bootstrapping loops to achieve an asymptotically reliable result. As mentioned in Section 2.4.4, the build-in function `gofCheckTime` allows estimating the necessary computation time for a given test, copula, dataset, and number of bootstrapping rounds. Since different machines may have highly varying computation times for tests, the function relies on a regression using the number of bootstrapping loops as the independent variable. To ensure that the linear model is a valid assumption, we investigated the case using the functions `gofKendallKS` and `gofKernel`; see Section 2.5.2 for the results.

Enabling parallelization of the bootstrapping is necessary for computationally demanding tests as `gofPIOSTn` where, e.g., the computation for a dataset of 500 observations and 1000 bootstrapping loops for the t -copula can take, depending on the engine, up to several hours. However, even for tests with faster computation time, parallelization is useful given the sample size and the number of bootstrapping loops is sufficiently high. This is shown in Table 2 in the form of a comparison between the computation times of five tests for five copulas without and with parallelization on four cores. The dataset contained $n = 500$ observations randomly generated from a bivariate standard normal distribution, and the number of bootstrapping loops was set to $M = 1000$.

Test	normal	t	clayton	gumbel	frank
<i>#processes = 1:</i>					
<code>gofKendallCvM</code>	200.33	530.22	166.85	249.86	167.81
<code>gofKendallKS</code>	115.43	450.50	100.69	172.26	88.83
<code>gofPIOSRn</code>	84.90	433.32	67.20	148.09	50.98
<code>gofRosenblattChisq</code>	75.79	431.02	73.09	143.94	55.97
<code>gofRosenblattGamma</code>	75.28	420.31	75.77	142.16	52.17
<i>#processes = 4:</i>					
<code>gofKendallCvM</code>	106.19	288.86	104.29	140.25	94.69
<code>gofKendallKS</code>	68.73	238.55	56.13	89.02	53.00
<code>gofPIOSRn</code>	48.81	235.35	47.37	83.71	33.66
<code>gofRosenblattChisq</code>	49.43	230.33	45.83	85.70	35.47
<code>gofRosenblattGamma</code>	47.45	234.50	48.41	82.04	37.32

Table 2: Computation times in seconds without and with parallelization using an Intel Core i7-4712MQ CPU with 2.3 GHz on a 64-Bit Windows 10 system.

5 Simulations

Empirical process of using the package

We would like to illustrate the power of the GoF tests with the use of the `gofCopula` package. In practice, one is often confronted with realizations of random variables for which an adequate copula model has to be found, as, e.g., in the two examples from the financial domain provided in Section 2.6. To illustrate the procedure, we focus in this Section on an easy replicable example. For this purpose, we start by simulating $n = 1000$ observations from a Clayton copula with Kendall's $\tau = 0.5$.

```
R> library("gofCopula")
R> param = iTau(copula = claytonCopula(), tau = 0.5)
R> n = 1000; set.seed(1)
R> x = rCopula(n = n, copula = claytonCopula(param = param))
```

To gain a better understanding of the data, Figure 2 shows the simulated data with different margins, reflecting the typical shape of the Clayton copula.

```
R> par(mfrow = c(1,2))
R> u = cbind(ecdf(x[,1])(x[,1]), ecdf(x[,2])(x[,2])) * n / (n + 1)
R> plot(u, col = "blue3", pch = 19, cex.lab = 1.25,
+       xlab = expression(u[1]), ylab = expression(u[2]))
R> plot(qnorm(u), col = "blue3", pch = 19, cex.lab = 1.25,
```

```
+ xlab = expression(x[1]), ylab = expression(x[2]))
R> par(mfrow = c(1,1))
```

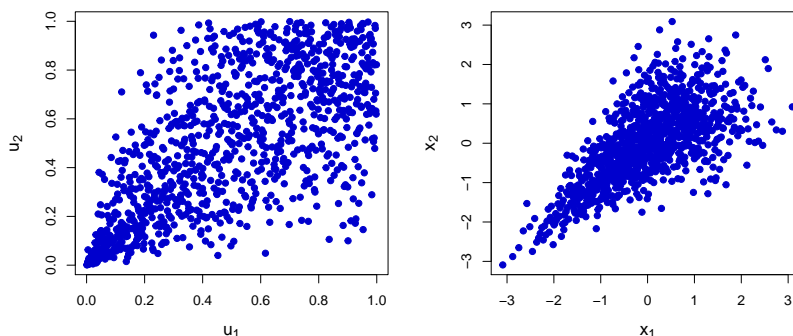


Figure 2: $n = 1000$ observations sampled from a Clayton copula with $\tau = 0.5$. Margins are transformed using ranks on the left plot and are standard normal on the right plot.

To make an adequate decision on which copula should be used in the respective modeling task, the GoF testing should involve more than looking at one or two test results and should consider a reasonable amount of potential copula models. We structure our procedure by testing for three groups of copulae separately: Elliptical, Archimedean, and extreme value (EV) copulae. In the function call, we select the FGM and Plackett copulae together with the EV category, although they do not belong to any of the three categories. Elliptical copulae include the normal and t -copula, while the Clayton, Gumbel, Frank, Joe, and AMH copulae are the Archimedean ones. Galambos, Husler-Reiss, Tawn, and t -EV belong to the EV category. Notice that this categorization could be modified, as, e.g., the Gumbel copula is also an EV copula. However, the given approach offers not only a logical structuring of the modeling task, but leads to using a close to maximal number of tests via only three function calls. The bootstrap parameters were set to $M = 100$ and $MJ = 1000$. As this task is computationally demanding, we set the argument `processes = 7` to speed up the calculation using parallelization on 7 cores. We use the default margins = "ranks" and set `seed.active = 10` for reproducibility.

```
R> cop_1 = gof(x = x, M = 100, MJ = 1000, processes = 7, seed.active = 10,
+           copula = c("normal", "t"))
R> cop_2 = gof(x = x, M = 100, MJ = 1000, processes = 7, seed.active = 10,
+           copula = c("clayton", "gumbel", "frank", "joe", "amh"))
R> cop_3 = gof(x = x, M = 100, MJ = 1000, processes = 7, seed.active = 10,
+           copula = c("galambos", "huslerReiss", "tawn", "tev", "fgm", "plackett"))
```

To evaluate the gained objects of class "gofCOP", one can manually inspect the resulting p -values and look closer at the performances and differences between the single tests and the corresponding hybrids. However, the easiest and most informative way is to visualize the p -values, which is done using the plot function.

```
R> plot(cop_1)
```

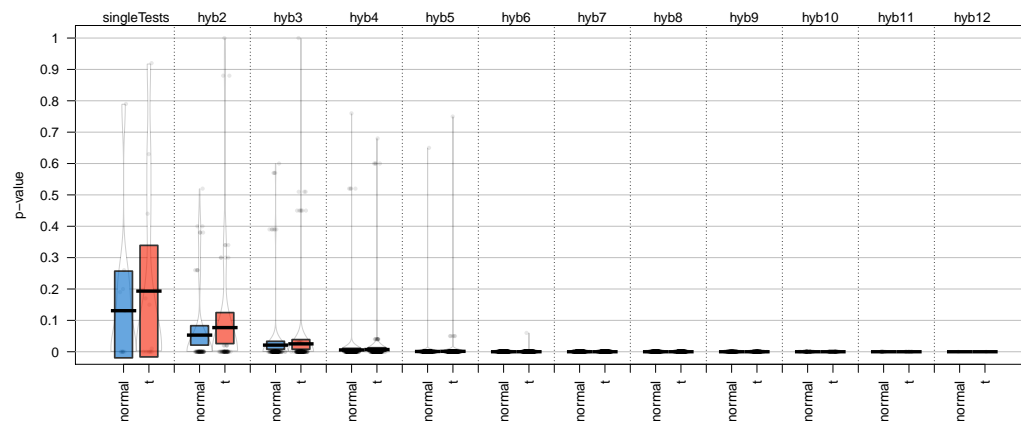


Figure 3: p -values of the hybrid tests for the data from Figure 2 for elliptical copulae.

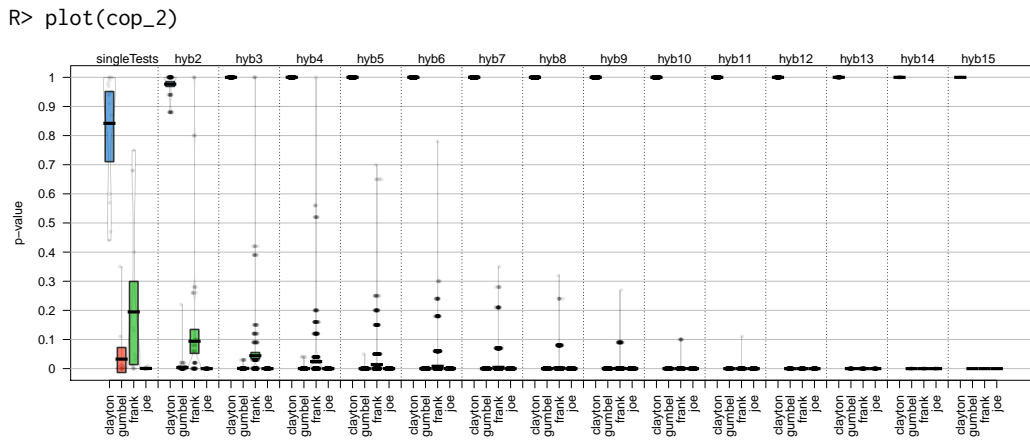


Figure 4: p -values of the hybrid tests for the data from Figure 2 for Archimedean copulae.

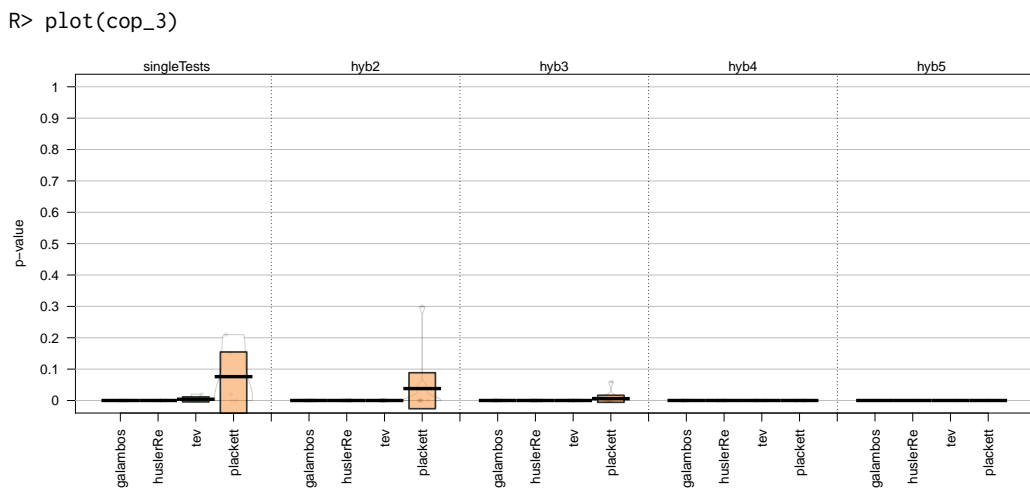


Figure 5: p -values of the hybrid tests for the data from Figure 2 for EV, FGM, and Plackett copulae.

Interpreting Figures 3, 4, and 5 clearly shows the ability of the tests to detect the true copula. The column `singleTests` in Figure 4 indicates that the Clayton copula is appropriate. The decision is supported by the higher-order hybrid tests, as all p -values except for the Clayton copula become 0, strongly rejecting the H_0 -hypothesis in these cases. Notice that similar to the introductory example in Section 2.4, the AMG, Tawn, and FGM are automatically excluded, which is why they do not appear in the plots. Having such a result at hand, the user can proceed with the modeling task with the selected copula.

Validating the model for the time estimation

In the next step, we validate the assumption of using a linear model for estimating the computation time in `gofCheckTime`. We have chosen the `gofKendallKS` test as a representative for the group of single bootstrapping tests and `gofKernel`, as the test having a double bootstrapping procedure. Both tests are available for all copulae in the bivariate case. For `gofKendallKS`, we measured the computation times for 12 copulae, varying numbers of bootstrap loops (M) and sample sizes (n) of the underlying dataset, which is simulated from a normal copula with Kendall's $\tau = 0.1$. This value is selected because it falls within the attainable interval of Kendall's τ for all copulae, see Table A.1. For `gofKernel`, we fixed $n = 100$ and investigated the situation for 12 copulae, different M , and different sample sizes MJ of the internal bootstrap. The results are shown in Figures 6, 7, and 8. The t -EV copula is not included in these illustrations due to its tremendous computation time, which can exceed the one of the t -Copula even for small sample sizes by a factor of 10 and higher. However, similar properties in the behavior of the computation time depending on the number of bootstrapping loops can be found. All calculations were performed without parallelization using an Intel Core i7-4712MQ CPU with 2.3 GHz on a 64-Bit Windows 10 system.

For the `gofKendallKS` test, the computation time increases linearly with the number of bootstrapping loops M , while the t -Copula is generally the most time-demanding of the considered copulae. This holds for all the analyzed sample sizes. A similar observation can be made for the `gofKernel` test. Here, a rapid increase in computation time is expected if both M and MJ increase. However, following Figures 7 and 8, this is not the case, and a linear dependency is justifiable. Therefore, the package implements for `gofKernel` a linear model with M and MJ being independent variables.

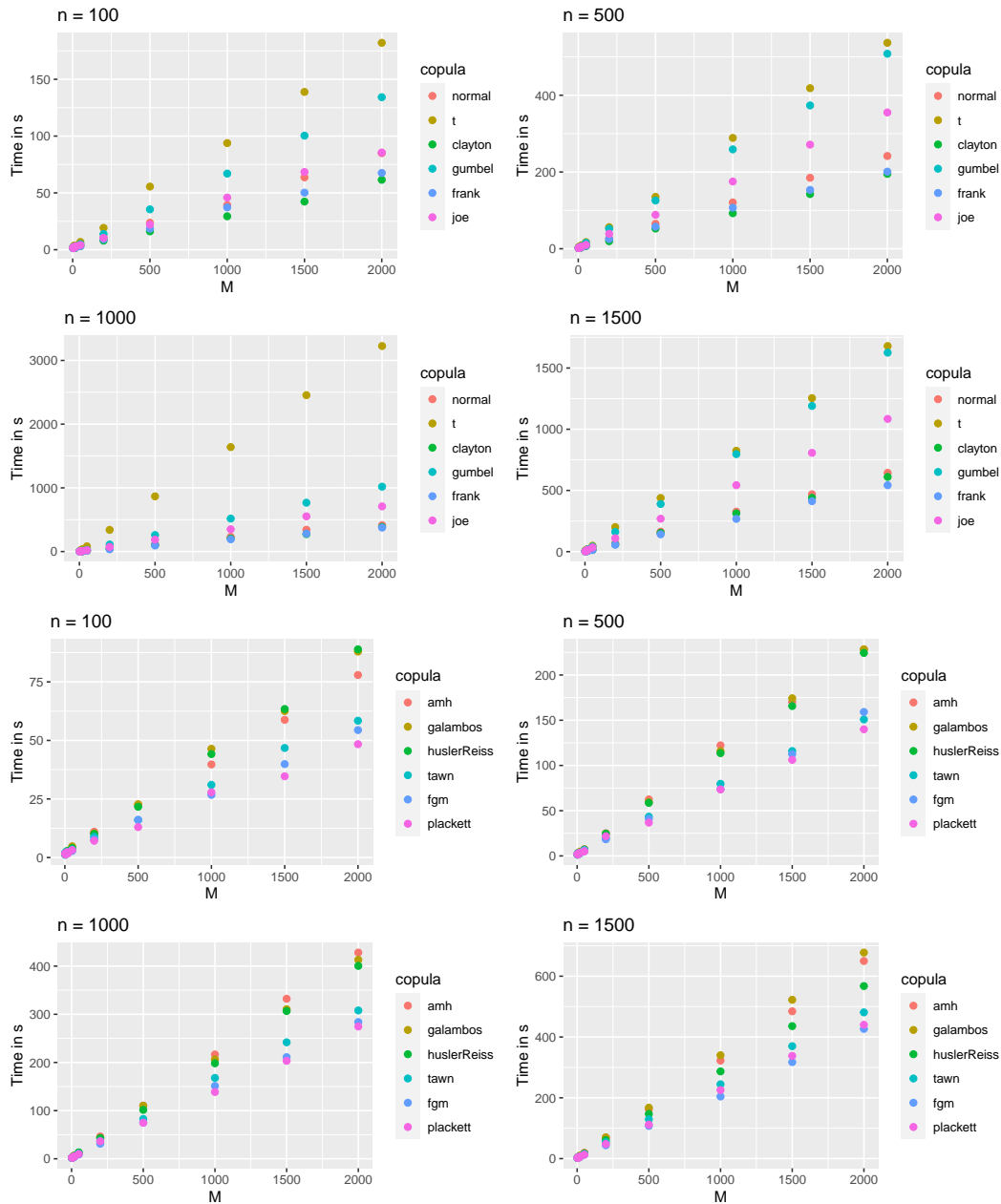


Figure 6: Computation times of `gofKendallKS` for different copulae, sample sizes n , and number of bootstrapping loops M .

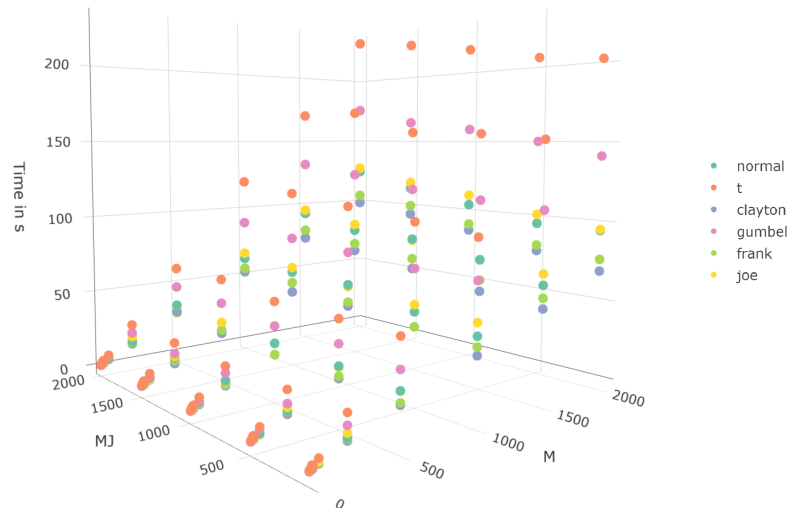


Figure 7: Computation times of `gofKernel` for different copulae, number of bootstrapping loops M , and internal bootstrap sample size MJ .

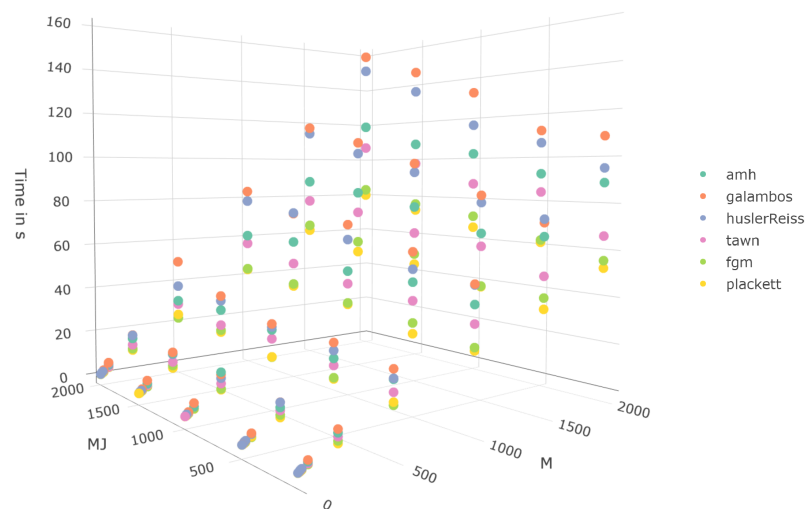


Figure 8: Computation times of `gofKernel` for different copulae, number of bootstrapping loops M , and internal bootstrap sample size MJ .

6 Application

Cryptocurrency market

We intend to demonstrate the functionality of the `gofCopula` package and show the empirical procedure as described in Section 2.5.1 on a real-world example from the market of cryptocurrencies. To account for the relevant steps in a realistic application study, we split the procedure into *Data Investigation* and *Goodness-of-Fit testing*.

Data investigation

We have chosen Bitcoin (BTC) and Litecoin (LTC) for our analysis. The objective is to detect which copula is appropriate to model the dependence structure between BTC-LTC and check whether the copula changes over the years. For that purpose, we use the volatility-adjusted log-returns of the currencies in the time span from 2015 to 2018. The volatility correction was performed by fitting a

GARCH(1,1) process to each time series for each year separately in order to extract their standardized residuals. These are included in the package as `CryptoCurrencies`, whereas each element of the list contains the data for a particular year. In order to gain a visual impression beforehand, we plotted the data with margins transformed to standard normal, leading to Figure 9. A strong dependency between both cryptocurrencies is visible, especially in the year 2018. Based on these residual diagrams, it is possible to take a guess which copula is the most adequate for the given situation. For 2015, one could possibly argue that the elliptical shape of a normal copula is present, while in 2016 and 2018, the shapes are more similar to the one of a t -Copula. Finally, for the year 2017, Figure 9 shows a comparable plot to Figure 2 from the simulated example in Section 2.5.1, indicating a Clayton copula might be present. However, these visual impressions are to a certain degree subjective and need to be backed up by the GoF tests. Ideally, the test results would match our plot-based guesses.

```
R> library("gofCopula")
R> data("CryptoCurrencies", package = "gofCopula")
R> par(mfrow = c(2,2))
R> years = as.character(2015:2018)

R> for(i in years){
+   x1 = CryptoCurrencies[[i]][,1]
+   x2 = CryptoCurrencies[[i]][,2]
+   n = length(x1)
+   plot(qnorm(cbind(ecdf(x1)(x1), ecdf(x2)(x2)) * n / (n + 1)), col = "blue3",
+         pch = 19, cex.lab = 1.25, main = i, xlab = "Bitcoin", ylab = "Litecoin"))
+ }
R> par(mfrow = c(1,1))
```

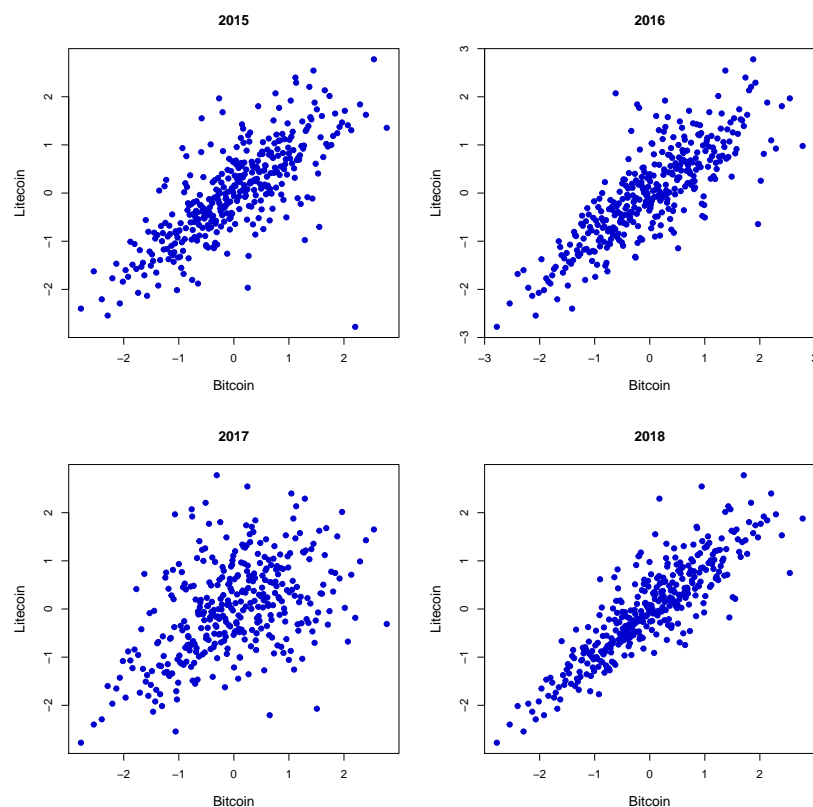


Figure 9: Residual plots for BTC-LTC with margins transformed to standard normal.

Goodness-of-fit testing

In this example, the focus in testing is on the most popular copula models in practice: normal, t , Clayton, Gumbel, and Frank copulae. To get the highest testing power, we include all tests, which are available for all five copulae. Thus, following Table 1, each test in the package except the ones based on the transformation for Archimedean copulae (see Section 2.3.4) is computed. Additionally, all

possible hybrid tests are considered. We use the function `gof` while setting the bootstrap parameters $M = 100$ and $MJ = 1000$. We specify the number of cores for the parallelization to `processes = 7`. For replicability, we set `seed.active = 1:101` and apply the non-parametric margin transformation by default.

```
R> copulae = c("normal", "t", "clayton", "gumbel", "frank")

R> BTC_LTC_15 = gof(x = CryptoCurrencies[["2015"]], copula = copulae, M = 100,
+                 MJ = 1000, processes = 7, seed.active = 1:101)
R> BTC_LTC_16 = gof(x = CryptoCurrencies[["2016"]], copula = copulae, M = 100,
+                 MJ = 1000, processes = 7, seed.active = 1:101)
R> BTC_LTC_17 = gof(x = CryptoCurrencies[["2017"]], copula = copulae, M = 100,
+                 MJ = 1000, processes = 7, seed.active = 1:101)
R> BTC_LTC_18 = gof(x = CryptoCurrencies[["2018"]], copula = copulae, M = 100,
+                 MJ = 1000, processes = 7, seed.active = 1:101)
```

After finishing the calculations, we proceed by plotting the received objects of class "gofCOP". For a detailed explanation about the information contained in the `gofCOP` pirateplots, please see Section 2.4.2 and Phillips (2017).

```
R> plot(BTC_LTC_15)
```

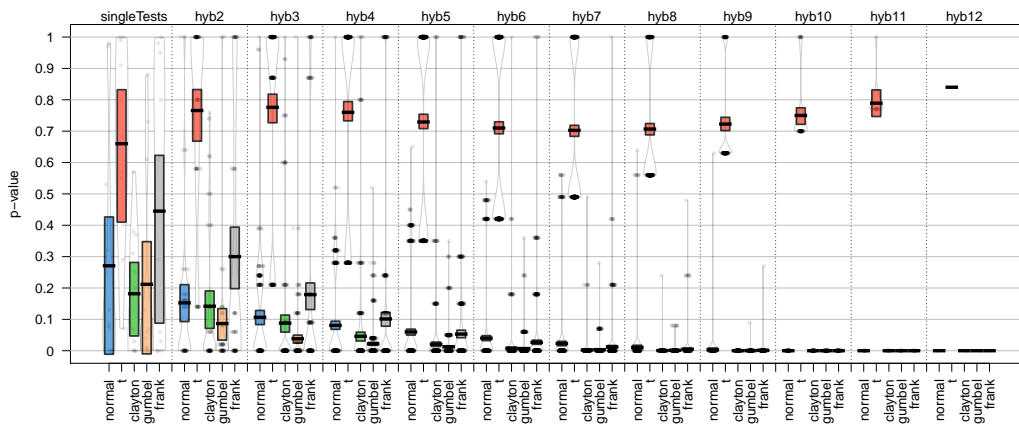


Figure 10: *p*-values of the single and hybrid tests for BTC-LTC in the year 2015.

```
R> plot(BTC_LTC_16)
```

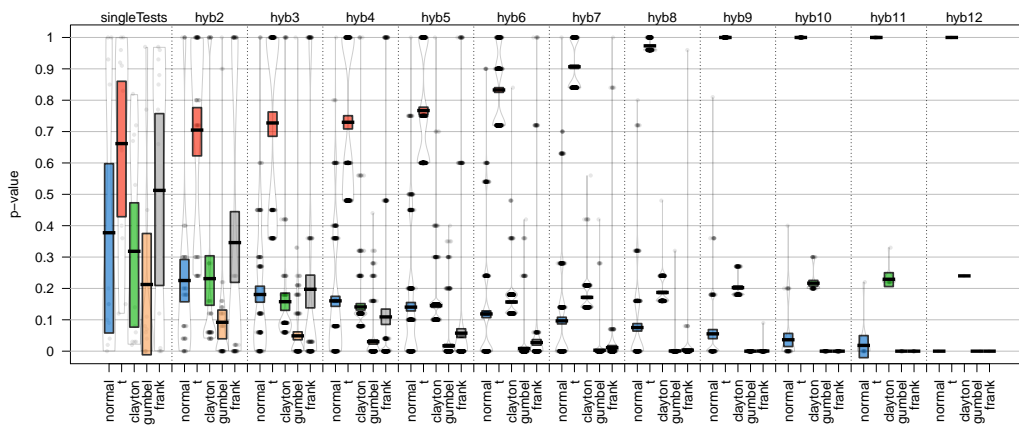


Figure 11: *p*-values of the single and hybrid tests for BTC-LTC in the year 2016.

R> plot(BTC_LTC_17)

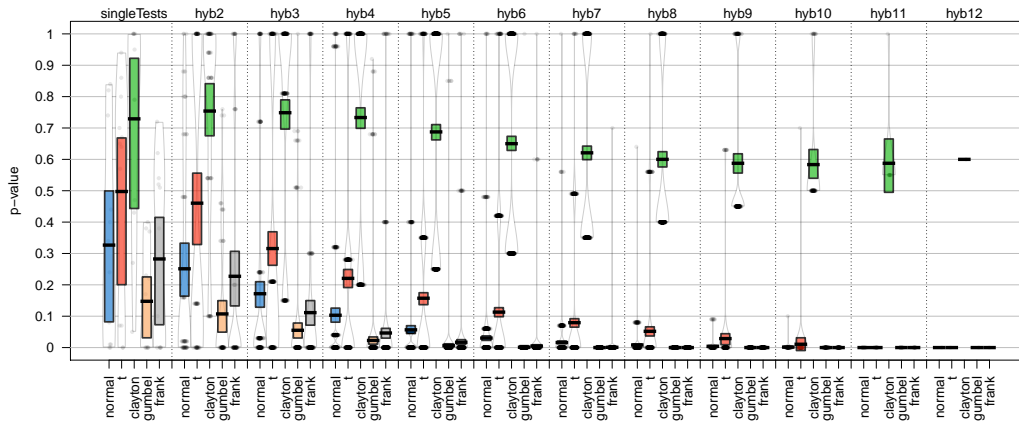


Figure 12: p -values of the single and hybrid tests for BTC-LTC in the year 2017.

R> plot(BTC_LTC_18)

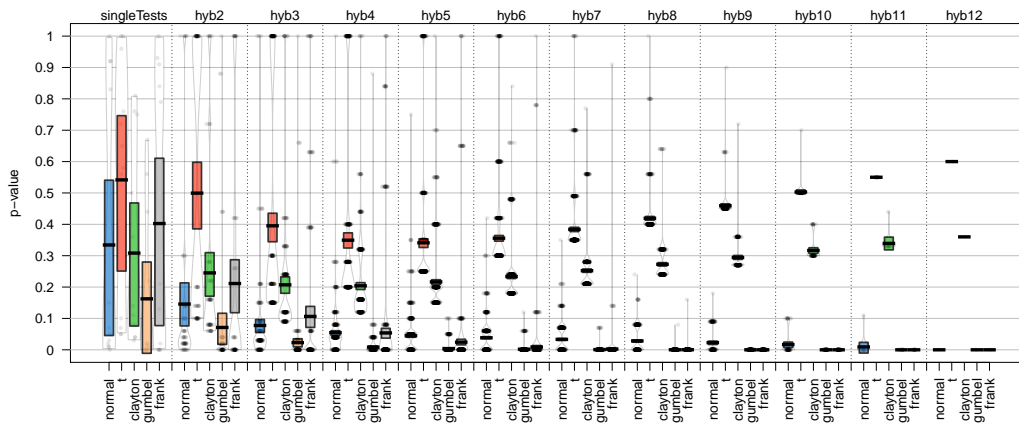


Figure 13: p -values of the single and hybrid tests for BTC-LTC in the year 2018.

Figures 10 to 13 show the resulting p -values in the form of "gofCOP"-plots for all the considered years. Following the usual approach in practice, we select the copula corresponding to the highest p -values. For the year 2015, we see that the t -copula is favored by the tests, as all remaining p -values become 0 with increasing hybrid testing size; see Figure 10. This rejects our initial visual guess that a normal copula might be appropriate. Continuing with 2016, we see our visual opinion solidified, as the plot suggests using a t -copula to capture the dependence structure. We can even see that the p -values converge to 1 for the t -copula when we consider the hybrid testing orders 9, 10, 11, and 12. Figure 12 is in line with our visual impression as well. We see that the tests favor a Clayton copula, while the other copula models are rejected by the higher-order hybrid tests. Finally, for 2018 Figure 13 gives for the t -copula the highest p -values, although the difference to the p -values of the Clayton copula is not too large. Therefore, in three out of four years, the results from **gofCopula** matched our visual impressions from the residual plots.

Summarizing, the following conclusions can be drawn from this analysis:

- Generally, the dominant copulae in describing the dependency between the volatility-adjusted log-returns of BTC-LTC is the t -copula. Following the test results, the year 2017 is an exception, as the dependence structure shifted towards a Clayton copula. This observation reflects the change in the market during the year 2017, as many investors got attracted by cryptocurrencies in this phase. Due to the developed hype, both the prices of BTC and LTC drastically increased, resulting in a modified underlying dependence structure between the two currencies.
- The hybrid tests are able to stabilize the results of the single tests, as they clearly selected for 2015, 2016, and 2018 the t -copula and in 2017 the Clayton copula. Therefore it is recommendable

to take hybrid tests into account in order to use the package adequately and get the highest testing power.

Stock return data

As a second real-world example, we analyze the volatility-adjusted stock log-returns of Citigroup (C) and the Bank of America (BoA) in the time span from 2004 to 2012. The procedure is, again, splitted into *Data Investigation* and *Goodness-of-Fit testing*.

Data investigation

This data was analyzed by [Zhang et al. \(2016\)](#), and we are expanding their procedure and consider the same copulae and tests as in the example in Section 2.6.1. The volatility correction was performed similarly in terms of fitting a GARCH(1,1) process, and the resulting data is included in `gofCopula` in the list `Banks`. Note that in this section, we focus on the years 2004 and 2007, while the results of the other years are given in Appendix B. We start by visualizing the residuals with margins transformed to standard normal.

```
R> library("gofCopula")
R> data("Banks", package = "gofCopula")
R> par(mfrow = c(1,2))

R> for(i in c("2004", "2007")){
+   x1 = Banks[[i]][,1]
+   x2 = Banks[[i]][,2]
+   n = length(x1)
+   plot(qnorm(cbind(ecdf(x1)(x1), ecdf(x2)(x2)) * n / (n + 1))), col = "blue3",
+        pch = 19, cex.lab = 1.25, main = i, xlab = "Citigroup", ylab = "Bank of America")
+ }
R> par(mfrow = c(1,1))
```

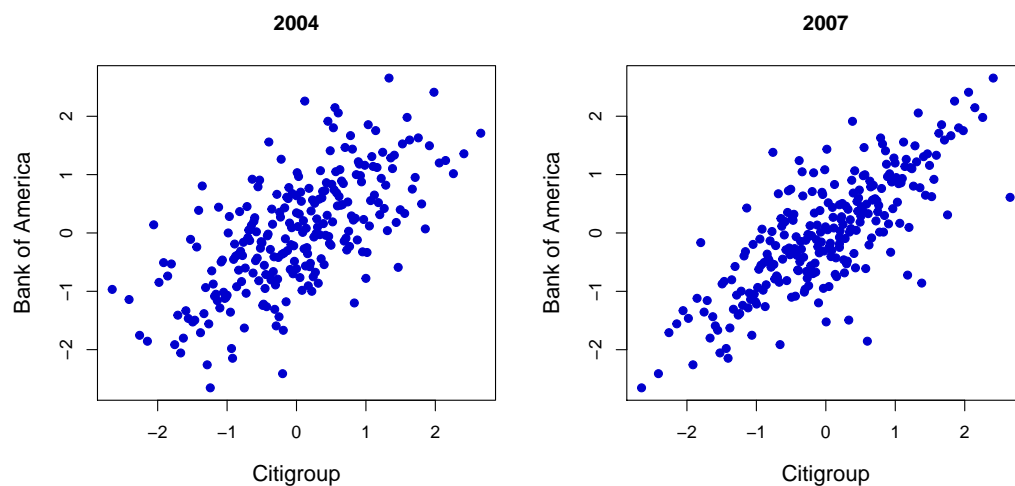


Figure 14: Residual plots for C/BoA in 2004 and 2007 with standard normal margins.

Analyzing the shape of the data in Figure 14 for 2004, one may argue that the elliptical normal copula is present, while in 2007, a t -copula is possibly more appropriate. To check these assumptions, we proceed with the GoF testing.

Goodness-of-fit testing

We set $M = 100$ and $MJ = 1000$ as bootstrap parameters, parallelize via `processes = 7`, and set `seed.active = 1:101` for reproducibility. Further, we implicitly keep the default `margins = "ranks"` to perform the margin transformation nonparametrically.

```
R> copulae = c("normal", "t", "clayton", "gumbel", "frank")

R> C_BoA_04 = gof(x = Banks[["2004"]], copula = copulae, M = 100, MJ = 1000,
+               processes = 7, seed.active = 1:101)
R> C_BoA_07 = gof(x = Banks[["2007"]], copula = copulae, M = 100, MJ = 1000,
+               processes = 7, seed.active = 1:101)
```

Following these calculations, we continue to plot the results, leading to Figures 15 and 16. For a detailed explanation about the information contained in the **gofCopula** pirateplots, please see Section 2.4.2 and Phillips (2017). Interpreting these "gofCOP"-plots of the p -values, the tests propose for 2004 indeed a normal copula (and a Frank one, which is radially symmetric), although a t -copula is a valid assumption as well. Compared to 2004, the p -values for the normal copula definitely decreased in 2007 and converged slowly to 0 with increasing hybrid testing size. The decision goes clearly in favor of the t -copula, which is in line with our original guess. Evaluating the results from Appendix B leads to similar conclusions as in Section 2.6.1. The hybrid tests are relatively stable and match in the majority of the cases the visual impressions from the residual plots. The proper copula seems to be the t -copula in most of the years, although in 2004 and 2009, the normal copula is a reasonable assumption. The hybrid tests are able to stabilize the selection of the copula.

```
R> plot(C_BoA_04)
```

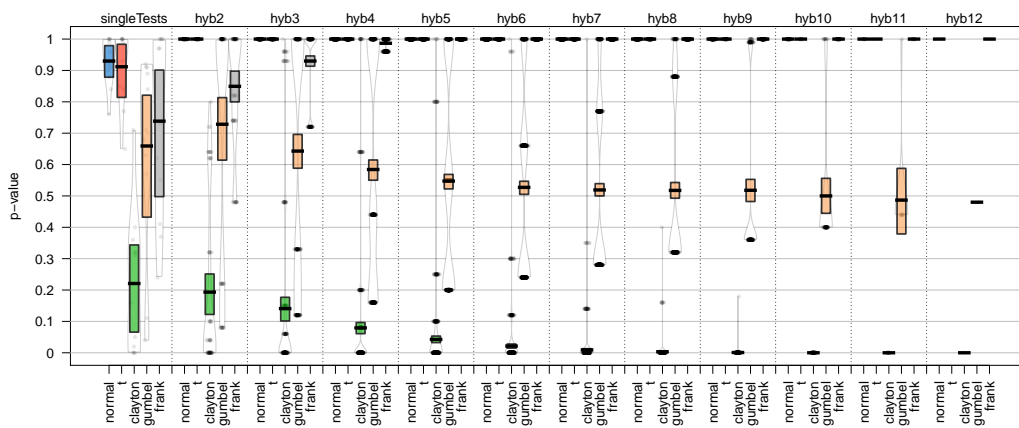


Figure 15: p -values of the C/BoA data for 2004. The column *hyb12* of the t -copula is empty, as the p -value of the test `gofWhite` could not be computed due to instability in the test statistics. For a detailed description of this phenomenon, see Nagler et al. (2019).

```
R> plot(C_BoA_07)
```

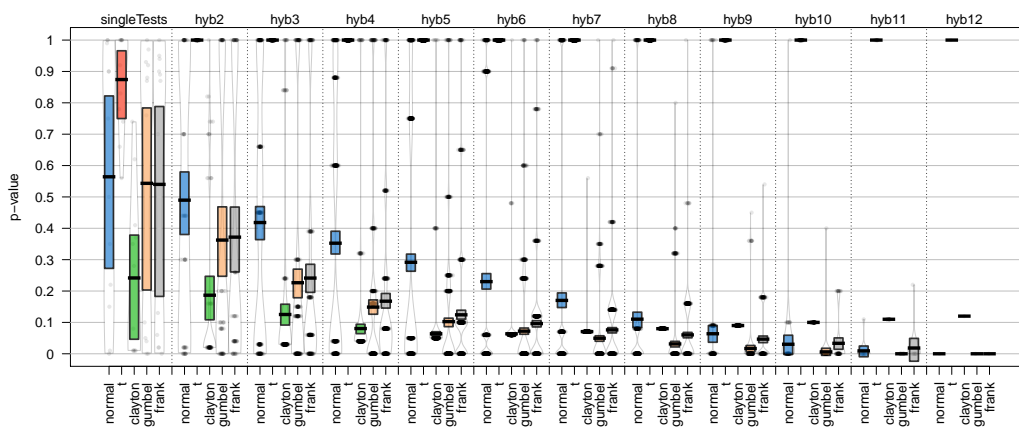


Figure 16: p -values of the C/BoA data for 2007.

7 Conclusion

This paper introduces a **gofCopula** package that provides maximum flexibility in performing statistical Goodness-of-Fit tests for copulae. The package provides an interface for 16 most popular GoF tests for 13 copulae with automatic estimation of margins via different techniques. The user is not limited to the implemented tests as self-defined test statistics functions can be easily embedded via a function provided in the package. As the computation of p -values relies on a parametric bootstrap, efficient and user-friendly parallelization is available. During the bootstrapping procedure, all tests inform the user about the progress of the calculations as well as the estimated time until the results are available. Additionally, **gofCopula** allows for the replication of said results. The package offers intelligible and interpretable visualization of the results of the hybrid tests that strengthen the overall test power. The flexibility and the usefulness of the tests are shown via a simulation and two empirical studies in economic sciences. In a nutshell, the broad range of tests, the comprehensive combination of methods, and an informative user-interface make **gofCopula** a fire-and-forget package providing flexibility in testing for the proper copula.

8 Acknowledgements

The authors are grateful to Shulin Zhang, Qian Zhou, Peter Song, and Sören Pannier for helpful discussions and to Oliver Scaillet for the code of the version of his test used in Zhang et al. (2016) that is being adapted in this package. Financial support from NUS FRC grant R-146-000-298-114 “Augmented machine learning and network analysis with applications to cryptocurrencies and blockchains” as well as CityU Start-Up Grant 7200680 “Textual Analysis in Digital Markets” is gratefully acknowledged by Simon Trimborn. Ostap Okhrin thankfully received financial support from RFBR, project number 20-04-60158.

Bibliography

- P. Barbe, C. Genest, K. Ghoudi, and B. Rémillard. On Kendall's process. *Journal of Multivariate Analysis*, 58:197–229, 1996. URL <https://doi.org/10.1006/jmva.1996.0048>. [p497]
- W. Breymann, A. Dias, and P. Embrechts. Dependence structures for multivariate high-frequency data in finance. *Quantitative Finance*, 1:1–14, 2003. URL <https://doi.org/10.1080/713666155>. [p498]
- S. X. Chen and T. Huang. Nonparametric estimation of copula functions for dependence modeling. *The Canadian Journal of Statistics*, 35(2):265–282, 2007. URL <https://doi.org/10.1002/cjs.5550350205>. [p496]
- X. Chen and Y. Fan. Estimation and model selection of semiparametric copula-based multivariate dynamic models under copula misspecification. *Journal of Econometrics*, 135(1–2):125–154, 2006. URL <https://doi.org/10.1016/j.jeconom.2005.07.027>. [p496]
- X. Chen, Y. Fan, and V. Tsyrennikov. Efficient estimation of semiparametric multivariate copula models. *Journal of the American Statistical Association*, 101(475):1228–1240, 2006. URL <https://doi.org/10.1198/016214506000000311>. [p496]
- G. Csárdi and R. FitzJohn. *progress: Terminal Progress Bars*, 2019. URL <https://CRAN.R-project.org/package=progress>. R package version 1.2.2. [p495]
- P. De Valpine. The common sense of p values. *Ecology*, 95(3):617–621, 2014. URL <https://doi.org/10.1890/13-1271.1>. [p493]
- S. Demarta and A. J. McNeil. The t -copula and related copulas. *International Statistical Review*, 73(1): 111–129, 2005. URL <https://doi.org/10.1111/j.1751-5823.2005.tb00254.x>. [p495, 521]
- J. Dobrić and F. Schmid. A goodness of fit test for copulas based on Rosenblatt's transformation. *Computational Statistics & Data Analysis*, 51(9):4633–4642, 2007. URL <https://doi.org/10.1016/j.csda.2006.08.012>. [p498]
- D. Dokuzoğlu and V. Purutçuoğlu. Comprehensive analyses of gaussian graphical model under different biological networks. *Acta Physica Polonica, A.*, 132(3), 2017. URL <https://doi.org/10.12693/APhysPolA.132.1106>. [p493]

- F. Durante and C. Sempi. Copula theory: An introduction. In P. Jaworski, F. Durante, W. K. Härdle, and T. Rychlik, editors, *Copula Theory and Its Applications*, pages 3–31. Springer, Berlin, Heidelberg, 2010. URL https://doi.org/10.1007/978-3-642-12465-5_1. [p493]
- Y. Fang and L. Madsen. Modified gaussian pseudo-copula: Applications in insurance and finance. *Insurance: Mathematics and Economics*, 53(1):292–301, 2013. URL <https://doi.org/10.1016/j.insmatheco.2013.05.009>. [p493]
- J.-D. Fermanian. Goodness-of-fit tests for copulas. *Journal of Multivariate Analysis*, 95(1):119–152, 2005. URL <https://doi.org/10.1016/j.jmva.2004.07.004>. [p497]
- J.-D. Fermanian and O. Scaillet. Nonparametric estimation of copulas for time series. *Journal of Risk*, 5: 25–54, 2003. URL <https://doi.org/10.21314/JOR.2003.082>. [p496]
- P. Gaensler and W. Stute. *Seminar on Empirical Processes*. Springer Basel AG, Boca Raton, 1987. URL <https://doi.org/10.1007/978-3-0348-6269-1>. [p496]
- C. Genest and J. Nešlehová. Copulas and copula models. *Wiley StatsRef: Statistics Reference Online*, 2014. URL <https://doi.org/10.1002/9781118445112.stat07523>. [p493]
- C. Genest and B. Rémillard. Validity of the parametric bootstrap for goodness-of-fit testing in semi-parametric models. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 6(44):1096–1127, 2008. URL <https://doi.org/10.1214/07-AIHP148>. [p497]
- C. Genest and L.-P. Rivest. Statistical inference procedures for bivariate Archimedean copulas. *Journal of the American Statistical Association*, 88(3):1034–1043, 1993. URL <https://doi.org/10.1080/01621459.1993.10476372>. [p495, 497]
- C. Genest, K. Ghoudi, and L.-P. Rivest. A semi-parametric estimation procedure of dependence parameters in multivariate families of distributions. *Biometrika*, 82(3):543–552, 1995. URL <https://doi.org/10.1093/biomet/82.3.543>. [p495, 496]
- C. Genest, J.-F. Quessy, and B. Rémillard. Goodness-of-fit procedures for copula models based on the probability integral transformation. *Scandinavian Journal of Statistics*, 33:337–366, 2006. URL <https://doi.org/10.1111/j.1467-9469.2006.00470.x>. [p497]
- C. Genest, B. Rémillard, and D. Beaudoin. Goodness-of-fit tests for copulas: A review and a power study. *Insurance: Mathematics and Economics*, 44:199–213, 2009. URL <https://doi.org/10.1016/j.insmatheco.2007.10.005>. [p493, 496, 497, 498, 500]
- T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. URL <https://doi.org/10.1198/01621450600001437>. [p493]
- M. Gomes, R. Radice, J. Camarena Brenes, and G. Marra. Copula selection models for non-gaussian outcomes that are missing not at random. *Statistics in medicine*, 38(3):480–496, 2019. URL <https://doi.org/10.1002/sim.7988>. [p493]
- J. Gröbner and O. Okhrin. Copulae: An overview and recent developments. *Wiley Interdisciplinary Reviews: Computational Statistics*, page e1557, 2021. URL <https://doi.org/10.1002/wics.1557>. [p493]
- C. Hering and M. Hofert. Goodness-of-fit tests for archimedean copulas in high dimensions. In K. Glau, M. Scherer, and R. Zagst, editors, *Innovations in Quantitative Risk Management*, pages 357–373, Cham, 2015. Springer International Publishing. URL https://doi.org/10.1007/978-3-319-09114-3_21. [p498]
- M. Hofert, I. Kojadinovic, M. Maechler, and J. Yan. *copula: Multivariate Dependence with Copulas*, 2020. URL <https://CRAN.R-project.org/package=copula>. R package version 1.0-0. [p493]
- K. Huang, L. Dai, M. Yao, Y. Fan, and X. Kong. Modelling dependence between traffic noise and traffic flow through an entropy-copula method. *Journal of Environmental Informatics*, 29(2), 2017. URL <https://doi.org/10.3808/jei.201500302>. [p493]
- W. Huang and A. Prokhorov. A goodness-of-fit test for copulas. *Econometric Reviews*, 33(7):751–771, 2014. URL <https://doi.org/10.1080/07474938.2012.690692>. [p499]
- H. Joe. *Multivariate Models and Dependence Concepts*. Chapman & Hall, London, 1997. [p495, 496]

- H. Joe. Asymptotic efficiency of the two-stage estimation method for copula-based models. *Journal of Multivariate Analysis*, 94(2):401–419, 2005. URL <https://doi.org/10.1016/j.jmva.2004.06.003>. [p496]
- H. Joe and D. Kurowicka. *Dependence Modeling: Vine Copula Handbook*. World Scientific, 2010. URL <https://doi.org/10.1142/7699>. [p493]
- M. Jouini and R. Clemen. Copula models for aggregating expert opinions. *Operations Research*, 3(44): 444–457, 1996. URL <https://doi.org/10.1287/opre.44.3.444>. [p497]
- S. Konigorski, Y. E. Yilmaz, and S. B. Bull. Bivariate genetic association analysis of systolic and diastolic blood pressure by copula models. *BMC Proceedings*, 8(1):S72, 2014. URL <https://doi.org/10.1186/1753-6561-8-S1-S72>. [p493]
- O. Kuss, A. Hoyer, and A. Solms. Meta-analysis for diagnostic accuracy studies: a new statistical model using beta-binomial distributions and bivariate copulas. *Statistics in medicine*, 33(1):17–30, 2014. URL <https://doi.org/10.1002/sim.5909>. [p493]
- Z. Liu, S. Guo, L. Xiong, and C.-Y. Xu. Hydrological uncertainty processor based on a copula function. *Hydrological sciences journal*, 63(1):74–86, 2018. URL <https://doi.org/10.1080/02626667.2017.1410278>. [p493]
- X. Ma, S. Luan, B. Du, and B. Yu. Spatial copula model for imputing traffic flow data from remote microwave sensors. *Sensors*, 17(10):2160, 2017. URL <https://doi.org/10.3390/s17102160>. [p493]
- F. Michiels and A. De Schepper. A copula test space model how to avoid the wrong copula choice. *Kybernetika*, 44(6):864–878, 2008. URL <http://hdl.handle.net/10338.dmlcz/135896>. [p521]
- T. Nagler, U. Schepsmeier, J. Stoeber, E. C. Brechmann, B. Graeler, and T. Erhardt. *VineCopula: Statistical Inference of Vine Copulas*, 2019. URL <https://CRAN.R-project.org/package=VineCopula>. R package version 2.3.0. [p493, 499, 516, 522, 523, 524]
- D. Oakes. Multivariate survival distributions. *Journal of Nonparametric Statistics*, 3(3-4):343–354, 1994. URL <https://doi.org/10.1080/10485259408832593>. [p496]
- D. H. Oh and A. J. Patton. Time-varying systemic risk: Evidence from a dynamic copula model of cds spreads. *Journal of Business & Economic Statistics*, 36(2):181–195, 2018. URL <https://doi.org/10.1080/07350015.2016.1177535>. [p493]
- N. Phillips. *yarr: A Companion to the e-Book “YaRrr!: The Pirate’s Guide to R”*, 2017. URL <https://CRAN.R-project.org/package=yarr>. R package version 0.1.5. [p495, 503, 513, 516]
- J.-D. F. D. Radulovic and M. Wegkamp. Weak convergence of empirical copula processes. *Bernoulli*, 10(5):847–860, 2004. URL <https://doi.org/10.3150/bj/1099579158>. [p496]
- B. Remillard and J.-F. Plante. *TwoCop: Nonparametric test of equality between two copulas*, 2012. URL <https://CRAN.R-project.org/package=TwoCop>. R package version 1.0. [p493]
- B. Remillard and O. Scaillet. Testing for equality between two copulas. *Journal of Multivariate Analysis*, 100:377–386, 2009. URL <https://doi.org/10.1016/j.jmva.2008.05.004>. [p493]
- M. Rosenblatt. Remarks on a multivariate transformation. *Annals of Mathematical Statistics*, 23(3): 470–472, 1952. URL <https://doi.org/10.1214/aoms/1177729394>. [p497, 498]
- I. D. L. Salvatierra and A. J. Patton. Dynamic copula models and high frequency data. *Journal of Empirical Finance*, 30:120–135, 2015. URL <https://doi.org/10.1016/j.jempfin.2014.11.008>. [p493]
- O. Scaillet. Kernel-based goodness-of-fit tests for copulas with fixed smoothing parameters. *Journal of Multivariate Analysis*, 98(3):533–543, 2007. URL <https://doi.org/10.1016/j.jmva.2006.05.006>. [p493, 499, 500]
- U. Schepsmeier. Efficient information based goodness-of-fit tests for vine copula models with fixed margins: A comprehensive review. *Journal of Multivariate Analysis*, 138:34–52, 2015. URL <https://doi.org/10.1016/j.jmva.2015.01.001>. [p499]
- P. Shi, X. Feng, and J.-P. Boucher. Multilevel modeling of insurance claims using copulas. *The Annals of Applied Statistics*, 10(2):834–863, 2016. URL <https://doi.org/10.1214/16-AOAS914>. [p493]

- J. H. Shih and T. A. Louis. Inferences on the association parameter in copula models for bivariate survival data. *Biometrics*, 51(4):1384–1399, 1995. URL <https://doi.org/10.2307/2533269>. [p496]
- A. Sklar. Fonctions de répartition à n dimension et leurs marges. *Publications de l'Institut de Statistique de l'Université de Paris*, 8:299–231, 1959. [p493]
- D. Valle and D. Kaplan. Quantifying the impacts of dams on riverine hydrology under non-stationary conditions using incomplete data and gaussian copula models. *Science of The Total Environment*, 677: 599–611, 2019. URL <https://doi.org/10.1016/j.scitotenv.2019.04.377>. [p493]
- W. Wang and M. Wells. Model selection and semiparametric inference for bivariate failure-time data. *Journal of the American Statistical Association*, 95(449):62–76, 2000. URL <https://doi.org/10.1080/01621459.2000.10473899>. [p497]
- H. White. Maximum likelihood estimation of misspecified models. *Econometrica: Journal of the Econometric Society*, 50:1–25, 1982. URL <https://doi.org/10.2307/1912526>. [p499, 500]
- F. Wu, E. Valdez, and M. Sherris. Simulating from exchangeable archimedean copulas. *Communications in Statistics—Simulation and Computation*, 36(5):1019–1034, 2007. URL <https://doi.org/10.1080/03610910701539781>. [p498]
- S. Zhang, O. Okhrin, Q. M. Zhou, and P. X.-K. Song. Goodness-of-fit test for specification of semiparametric copula dependence models. *Journal of Econometrics*, 193(1):215–233, 2016. URL <https://doi.org/10.1016/j.jeconom.2016.02.017>. [p493, 494, 500, 515, 517]

1 Appendix A

Table A.1 contains parameter ranges, the bivariate cumulative distribution function (CDF), and possible values of Kendall’s τ for the copulae available in **gofCopula**.

Copula	$\theta \in$	$C(u_1, u_2)$	$\tau \in$
Normal	$[-1, 1]$	$\int_{-\infty}^{\Phi^{-1}(u_1)} \int_{-\infty}^{\Phi^{-1}(u_2)} \frac{1}{2\pi\sqrt{1-\theta^2}} \exp\left\{\frac{2\theta st - s^2 - t^2}{2(1-\theta^2)}\right\} ds dt$	$[-1, 1]$
t	$\begin{matrix} [-1, 1] \\ \nu > 0 \end{matrix}$	$\int_{-\infty}^{t_v^{-1}(u_1)} \int_{-\infty}^{t_v^{-1}(u_2)} \frac{1}{2\pi\sqrt{1-\theta^2}} \left\{1 + \frac{s^2 + t^2 - 2\theta st}{\nu(1-\theta^2)}\right\}^{-\frac{\nu+2}{2}} ds dt$	$[-1, 1]$
Clayton	$[-1, \infty) \setminus \{0\}$	$\left\{\max(u_1^{-\theta} + u_2^{-\theta} - 1, 0)\right\}^{-\frac{1}{\theta}}$	$[-1, 1]$
Gumbel	$[1, \infty)$	$\exp\left[-\left\{(-\log u_1)^{\frac{1}{\theta}} + (-\log u_2)^{\frac{1}{\theta}}\right\}^{\theta}\right]$	$[0, 1]$
Frank	$(-\infty, \infty) \setminus \{0\}$	$-\frac{1}{\theta} \log\left[1 + \frac{\{\exp(-\theta u_1) - 1\}\{\exp(-\theta u_2) - 1\}}{\exp(-\theta) - 1}\right]$	$[-1, 1]$
Joe	$[1, \infty)$	$1 - \left\{(1 - u_1)^{\theta} + (1 - u_2)^{\theta} - (1 - u_1)^{\theta}(1 - u_2)^{\theta}\right\}^{\frac{1}{\theta}}$	$[0, 1]$
AMH	$[-1, 1]$	$\frac{u_1 u_2}{1 - \theta(1 - u_1)(1 - u_2)}$	$\left[\frac{5 - 8 \log 2}{3}, \frac{1}{3}\right]$ $\approx [-0.1817, 0.3333]$
Galambos	$[0, \infty)$	$u_1 u_2 \exp\left[\left\{(-\log u_1)^{-\theta} + (-\log u_2)^{-\theta}\right\}^{-\frac{1}{\theta}}\right]$	$[0, 1]$
Husler-Reiss	$[0, \infty)$	$\exp\left\{\log(u_1)\Phi\left(\frac{1}{\theta} + \frac{1}{2}\theta \log \frac{\log u_1}{\log u_2}\right) + \log(u_2)\Phi\left(\frac{1}{\theta} + \frac{1}{2}\theta \log \frac{\log u_2}{\log u_1}\right)\right\}$	$[0, 1]$
Tawn	$[0, 1]$	$u_1 u_2 \exp\left(-\theta \frac{\log u_1 \log u_2}{\log u_1 + \log u_2}\right)$	$\left[0, \frac{8 \arctan(\sqrt{\frac{1}{3}})}{\sqrt{3}} - 2\right]$ $\approx [0, 0.4184]$
t -EV	$\begin{matrix} [-1, 1] \\ \nu > 0 \end{matrix}$	see Demarta and McNeil (2005)	$[0, 1]$
FGM	$[-1, 1]$	$u_1 u_2 + u_1 u_2 \theta(1 - u_1)(1 - u_2)$	$[-\frac{2}{9}, \frac{2}{9}]$
Plackett	$(0, \infty)$	$\frac{\{1 + (\theta - 1)(u_1 + u_2)\} - \sqrt{\{1 + (\theta - 1)(u_1 + u_2)\}^2 - 4u_1 u_2 \theta(\theta - 1)}}{2(\theta - 1)}$	$[-1, 1]$

Table A.1: This table is mainly based on Michiels and De Schepper (2008). AMH abbreviates Ali-Mikhail-Haq. FGM stands for Farlie-Gumbel-Morgenstern. Φ is the CDF of the univariate standard normal distribution and Φ^{-1} its inverse. t_v^{-1} is the inverse CDF of the univariate t -distribution with ν degrees of freedom. The expression of the CDF for the t -EV is complex due to the construction via a Pickands dependence function, which is why we do not explicitly list it. The given parameterization of the tawn copula is based on the one implemented in the package **copula**.

2 Appendix B

This section is devoted to the results of Section 2.6.2.

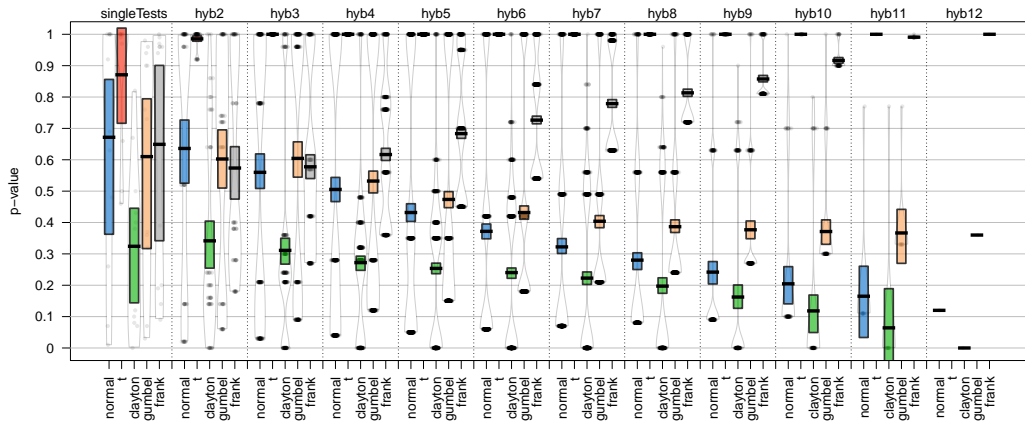


Figure B.1: *p*-values of the C/BoA data for 2005. The column *hyb12* of the *t*-copula is empty, as the *p*-value of the test *gofWhi* te could not be computed due to instability in the test statistics. For a detailed description of this phenomenon, see Nagler et al. (2019).

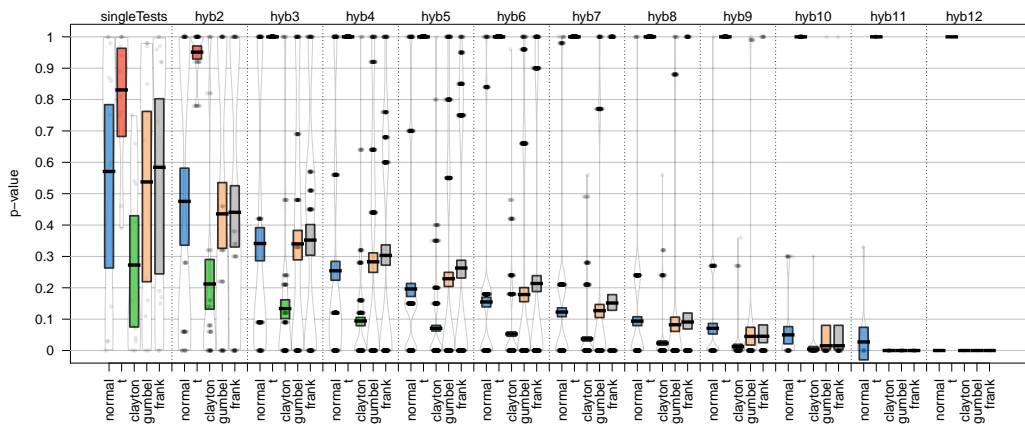


Figure B.2: *p*-values of the C/BoA data for 2006.

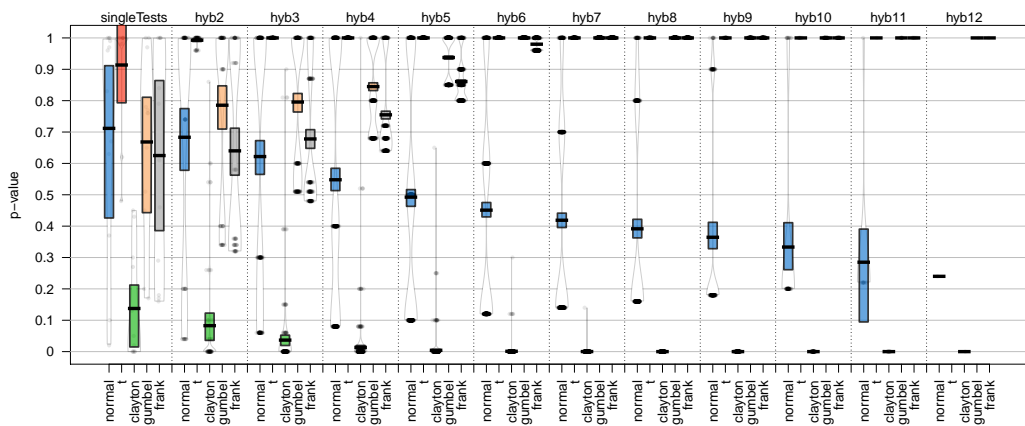


Figure B.3: *p*-values of the C/BoA data for 2008. The column *hyb12* of the *t*-copula is empty, as the *p*-value of the test *gofWhi* te could not be computed due to instability in the test statistics. For a detailed description of this phenomenon, see Nagler et al. (2019).

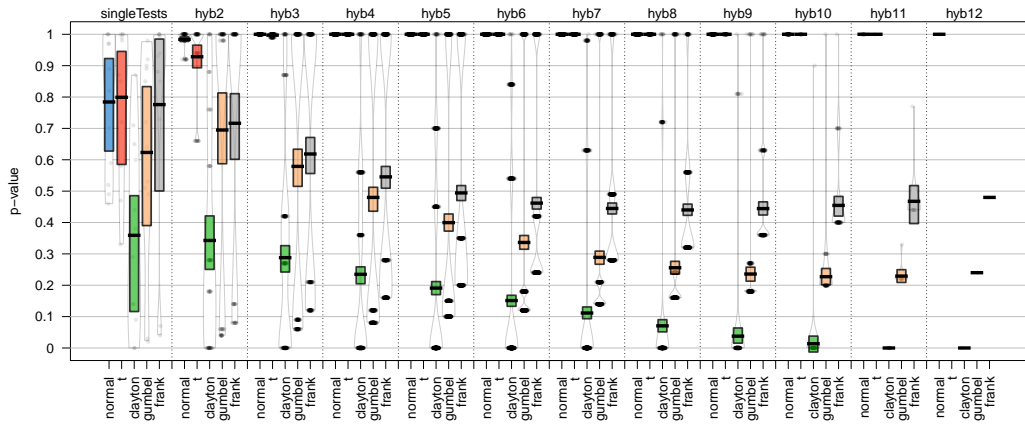


Figure B.4: p -values of the C/BoA data for 2009. The column *hyb12* of the t -copula is empty, as the p -value of the test *gofWhi* te could not be computed due to instability in the test statistics. For a detailed description of this phenomenon, see Nagler et al. (2019).

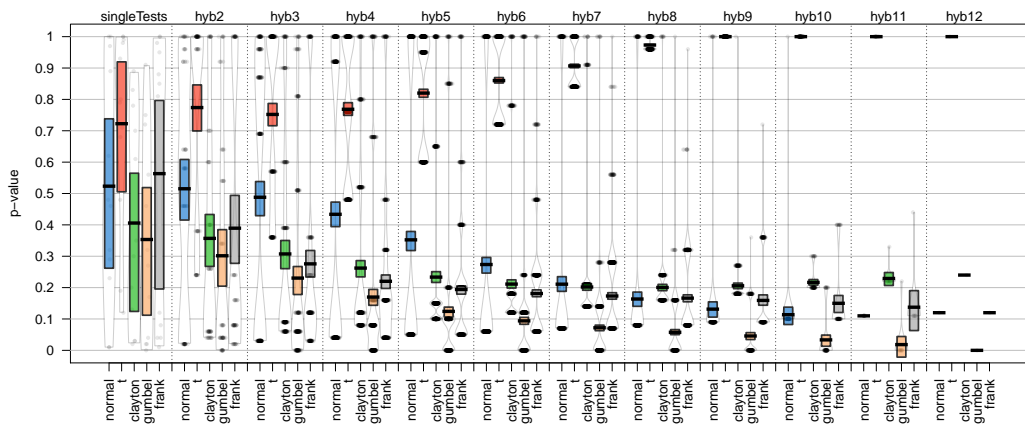


Figure B.5: p -values of the C/BoA data for 2010.

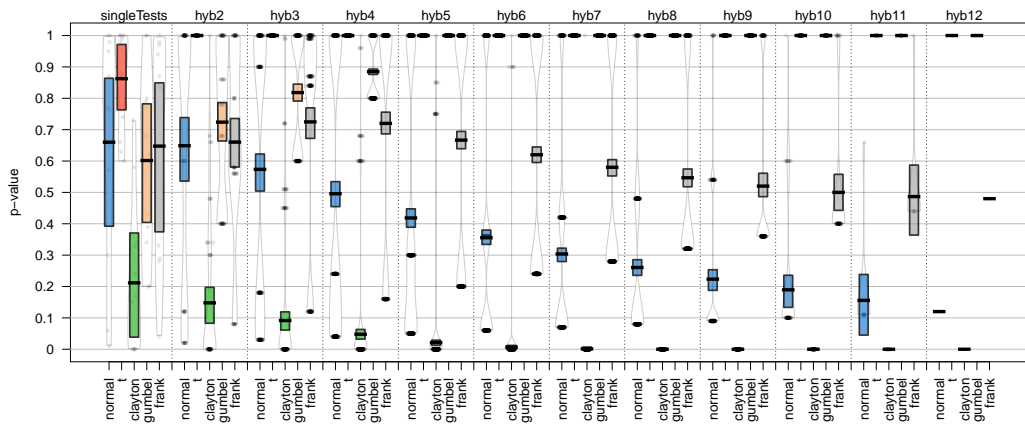


Figure B.6: p -values of the C/BoA data for 2011.

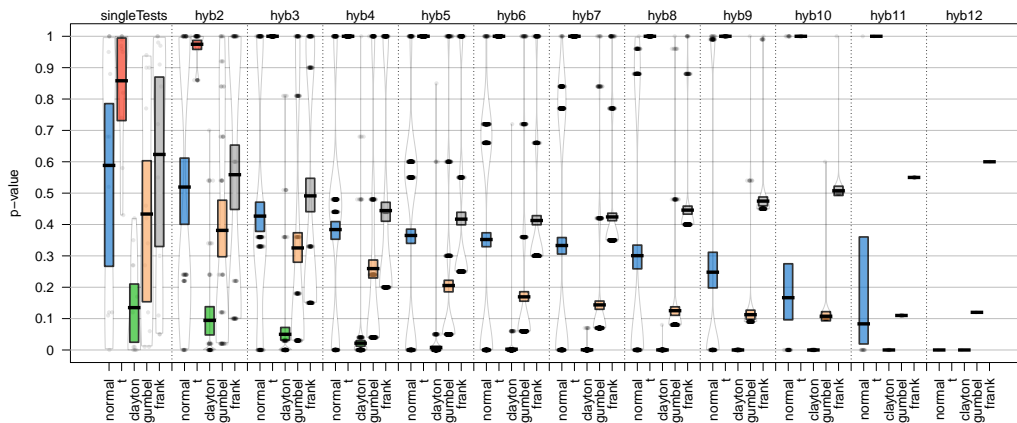


Figure B.7: p -values of the C/BoA data for 2012. The column $hyb12$ of the t -copula is empty, as the p -value of the test $gofWhi$ te could not be computed due to instability in the test statistics. For a detailed description of this phenomenon, see Nagler et al. (2019).

Ostap Okhrin
 Chair of Econometrics and Statistics, esp. in Traffic Sciences
 Institute of Transportation Economics
 Faculty of Transportation
 Technische Universität Dresden
 Würzburger Street 35, 01187 Dresden
 Germany
ostap.okhrin@tu-dresden.de

Simon Trimborn
 Department of Management Sciences
 and School of Data Science
 City University of Hong Kong
 7-268, Lau Ming Wai Academic Building
 Hong Kong
trimborn.econometrics@gmail.com

Martin Waltz
 Chair of Econometrics and Statistics, esp. in Traffic Sciences
 Institute of Transportation Economics
 Faculty of Transportation
 Technische Universität Dresden
 Würzburger Street 35, 01187 Dresden
 Germany
martin.waltz@tu-dresden.de