Université de Montréal

A consistent test of independence between random vectors

par

Guillaume Boglioni Beaulieu

Département de mathématiques et de statistique
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en statistique

24 novembre 2016

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé

A consistent test of independence between random vectors

présenté par

Guillaume Boglioni Beaulieu

a été évalué par un jury composé des personnes suivantes :

Martin Bilodeau
(président-rapporteur)

Pierre Lafaye de Micheaux
(directeur de recherche)

Benjamin Avanzi
(codirecteur)

Bernard Wong
(codirecteur)

Maciej Augustyniak
(membre du jury)

Mémoire accepté le
20 décembre 2016

SUMMARY

Testing for independence between random vectors is an important question in statistics. Because there is an infinite number of ways by which a random quantity X can be dependent of another random quantity Y, it is not a trivial question. It has been found that classical tests such has Spearman [33],Wilks [40], Kendall [18] or Puri and Sen [24] are ineffective to detect many forms of dependence. Recent, significant results on the topic include Székely et al. [35], Gretton et al. [14] or Heller et al. [15]. However, most of the available tests can only detect dependence between two random quantities. Because pairwise independence does not guarantee mutual independence, techniques testing the hypothesis of mutual independence between *any* number of random quantities are required. In this research we propose a non-parametric and universally consistent test of independence, applicable to any number of random vectors of any size.

Precisely, we extend the procedure described in Heller et al. [15] in two ways. Firstly, we propose to use the ranks of the observations instead of the observations themselves. Secondly, we extend their method to test for independence between any number of random vectors. As the distribution of our test statistic is not known, a permutation method is used to compute $p-$values. Then, simulations are performed to obtain the power of the test. We compare the power of our new test to that of other tests, namely those in Genest and Rémillard [10], Gretton et al. [14], Székely et al. [34], Beran et al. [3] and Heller et al. [15]. Examples featuring random variables and random vectors are considered. For many examples investigated we find that our new test has similar or better power than that of the other tests. In particular, when the random variables are Cauchy, our new test outperforms the others. In the case of strictly discrete random vectors, we present a proof that our test is universally consistent.

Keywords: Independence test, multivariate data, random vectors

SOMMAIRE

Tester l'indépendance entre plusieurs vecteurs aléatoires est une question importante en statistique. Puisqu'il y a une infinité de manières par lesquelles une quantité aléatoire X peut dépendre d'une autre quantité aléatoire Y, ce n'est pas une question triviale, et plusieurs tests "classiques" comme Spearman [33], Wilks [40], Kendall [18] ou Puri and Sen [24] sont inefficaces pour détecter plusieurs formes de dépendance. De significatifs progrès dans ce domaine ont été réalisés récemment, par exemple dans Székely et al. [34], Gretton et al. [14] ou Heller et al. [15]. Cela dit, la majorité des tests disponibles détectent l'indépendance entre deux quantités aléatoires uniquement. L'indépendance par paires ne garantissant pas l'indépendance mutuelle, il est pertinent de développer des méthodes testant l'hypothèse d'indépendance mutuelle entre n'importe quel nombre de variables. Dans cette recherche nous proposons un test non-paramétrique et toujours convergent, applicable à un nombre quelconque de vecteurs aléatoires.

Précisément, nous étendons la méthode décrite dans Heller et al. [15] de deux manières. Premièrement, nous proposons d'appliquer leur test aux rangs des observations, plutôt qu'aux observations elles-mêmes. Ensuite, nous étendons leur méthode pour qu'elle puisse tester l'indépendance entre un nombre quelconque de vecteurs. La distribution de notre statistique de test étant inconnue, nous utilisons une méthode de permutations pour calculer sa valeur-p. Des simulations sont menées pour obtenir la puissance du test, que nous comparons à celles d'autres test décrits dans Genest and Rémillard [10], Gretton et al. [14], Székely et al. [34], Beran et al. [3] et Heller et al. [15]. Nous investiguons divers exemples et dans plusieurs de ceux-ci la puissance de notre test est meilleure que celle des autres tests. En particulier, lorsque les variables aléatoires sont Cauchy notre test performe bien mieux que les autres. Pour le cas de vecteurs aléatoires strictement discrets, nous présentons une preuve que notre test est toujours convergent.

Keywords: Test d'indépendance, données multivariées, vecteurs aléatoires

CONTENTS

ACKNOWLEDGMENTS

First I must thank my supervisor Pierre Lafaye de Micheaux as well as my co-supervisors Benjamin Avanzi and Bernard Wong. Their support and always on-point suggestions had a huge positive impact on the quality of this thesis.

I thank my friends and fellow research students, whose company made the completion of this degree a pleasant journey. Thanks to my UdeM companions Francis Lafontaine, David Berger, Philippe Gagnon, Michaël Lalancette and Alexandre Carbonneau. Thanks also to my UNSW companions Vincent Tu, Nicolay Gudkov, Alan Xian, Anh Vu and Xinda Yang.

I thank my teammates from University of Montreal's Athletics Club. My experience in university would not have been the same without them. Special thanks to my coach Éric Janvier, whose devotion to his athletes and to the team has no limits.

I thank my mother, whose unconditional love and support will always amaze me. I thank my father, who taught me almost everything. I like to believe he would have been proud.

Chapter 1

INTRODUCTION

1.1. General background

The concept of *independence* between events or random variables plays a crucial role in probability and statistics. It is introduced quickly in any basic undergraduate course of these disciplines. An intuitive formulation of the concept is stated in Resnick [27] as follows:

> Independence is a basic property of events and random variables in a probability model. Its intuitive appeal stems from the easily envisioned property that the occurrence or non-occurrence of an event has no effect on our estimate of the probability that an independent event will or will not occur.

Another way to phrase this might be to say that two events are independent when knowledge about one of them gives absolutely no information about the other. A more formal definition of *independent random variables* will be given later on, as this research project is primarily interested in answering the question *how do we detect dependence?*

1.2. Research aim

The objective of this thesis is to build a statistical procedure to test the hypothesis of mutual independence between a collection of random vectors. We propose a test that is non-parametric (i.e. distribution free), universally consistent (i.e. with a sample size large enough the test will detect any form of dependence) and applicable to any number of random vectors of any size. To do so, our starting point is the procedure proposed in Heller et al. [15] which is consistent to detect dependence between two random vectors X and Y. To summarize, the present research offers two contributions.

First, in the bivariate setting (i.e. testing the independence of two random vectors X and Y) we propose a simple modification of Heller et al. [15] that we believe is best suited in many situations. Secondly, we extend the methodology of Heller et al. [15] to test for multivariate independence.

To avoid ambiguity let us state right away that by a 'multivariate' independence testing we mean detecting dependence between more than two random variables or vectors. Hence, although others might call testing the independence between two vectors a multivariate procedure, this is not what we intend when we use the term multivariate.

1.3. Research motivation

To quote Sen and Srivastava [30], "perception of relationships is the cornerstone of civilization. By understanding how certain phenomena depend on others we learn to predict the consequences of our actions and to manipulate our environment". This is well-said, and indeed as statisticians we wish to understand the relations dictating the real-life phenomena that surrounds us. At the very base of any statistical model there is the notion that information about one or many variables contains some information about one or many other variables.

Hence we will say, perhaps in a lack of humility, that detecting dependence is the very basic problem in statistics. Before we can *explain* the relation between two variables, say X and Y, we want to confidently answer the more fundamental question *is there a relation between X and Y?* This is because any attempt at modeling would be a waste of time if they are in fact independent.

Now, this problem is not new. If this is an old problem, and very simple to write down, why is there so much literature still being produced on the topic? One answer might be that there is an infinite number of ways by which a random quantity X can be dependent of another random quantity Y. We aim at developing statistical tests valid against any departure from independence (i.e. universally consistent), but this is not at trivial task. It has been found, see for instance Székely et al. [34], that classical tests such has Spearman [33], Wilks [40], Kendall [18] or Puri and Sen [24], although vastly used in practice, are ineffective to detect many forms of dependence.

1.3.1. An example of non-monotone dependence

To illustrate this, we consider a first example with a dataset taken from the *The World Factbook*, a publication from the CIA (Central Intelligence Agency)[1]. This dataset contains the 2015 birth rates and mortality rates from 145 countries with a population over 2 millions. Note these are the 'crude' rates, meaning aggregated at all ages. The scatter plot of birth rate against mortality rate is displayed on Figure 1.1.

[1]https://www.cia.gov/library/publications/the-world-factbook/geos/xx.html

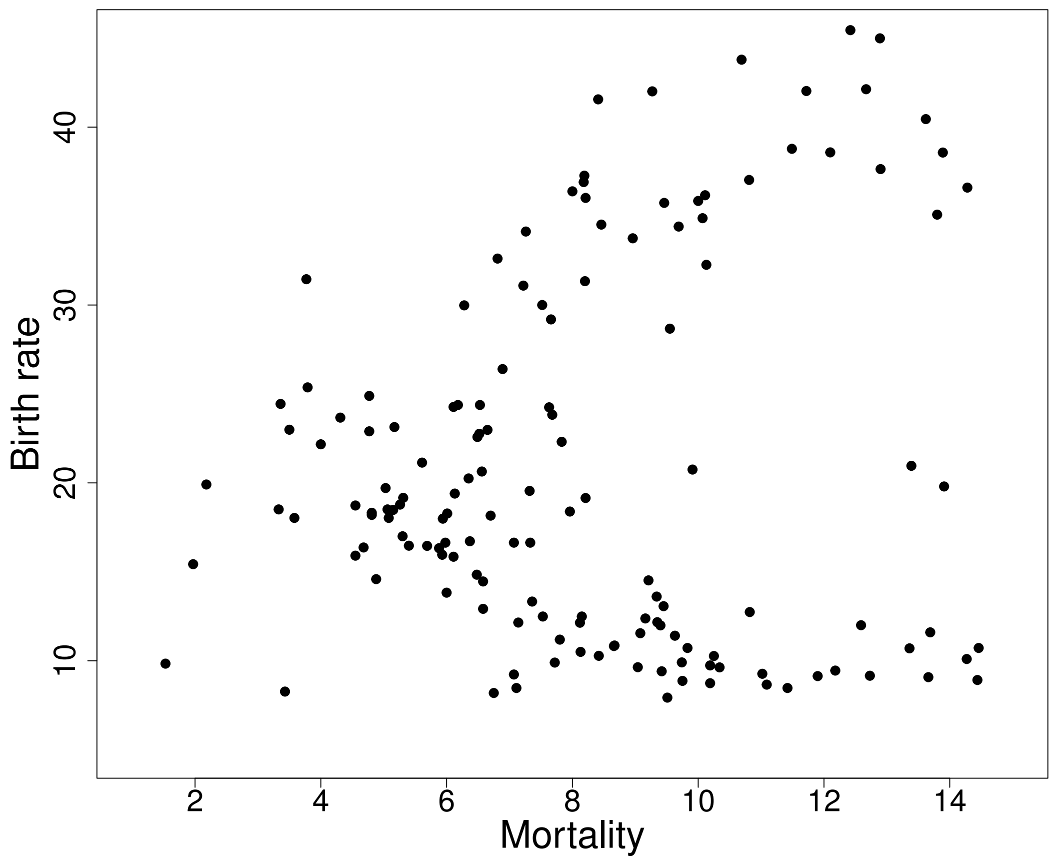

FIG. 1.1. Birth rate against mortality in 145 countries

We notice quite clearly a pattern between the two variables, a kind of "C" shape. Because any recognizable pattern between the variables reveals dependence, we would expect a consistent test to find it. However, if we compute Pearson's correlation for this dataset we obtain a value of 0.139, and a p-value (testing the hypothesis that the correlation is zero) of 0.095. Therefore, with the conventional level of significance of 5% we do not conclude that the coefficient of correlation is different than zero. The same goes for Spearman's rho that has a value of -0.035 (p-value of 0.675) and Kendall's tau that has a value of -0.032 (p-value of 0.574).

Hence these classical and vastly used methods fail to detect a dependence that is quite clear at the sight of the scatter plot. This is so because the dependence displayed here is not linear, nor is it monotone. To detect dependence in such a context, we need more recent methods. For instance, distance covariance by Székely et al. [35], or the HHG test by Heller et al. [15] (named after the authors

Heller, Heller and Gorfine) both result in a p-value < 0.001, hence rejecting independence. The test we propose in the present research also yields a p-value < 0.001.

To make this example slightly more complete we make one final comment. Although we know very little about demography, it would probably be very simplistic to investigate birth rate as a function of mortality only. If we separate the 145 countries in two groups, based on their GDP per capita (one group containing the 33% poorest countries, and the other containing the remaining richest countries), we obtain a much clearer picture of the situation, see Figure 1.2. We can identify two very different trends in those groups. Hence, here the interaction with a third variable is relevant to understand the relation between birth rate and mortality.

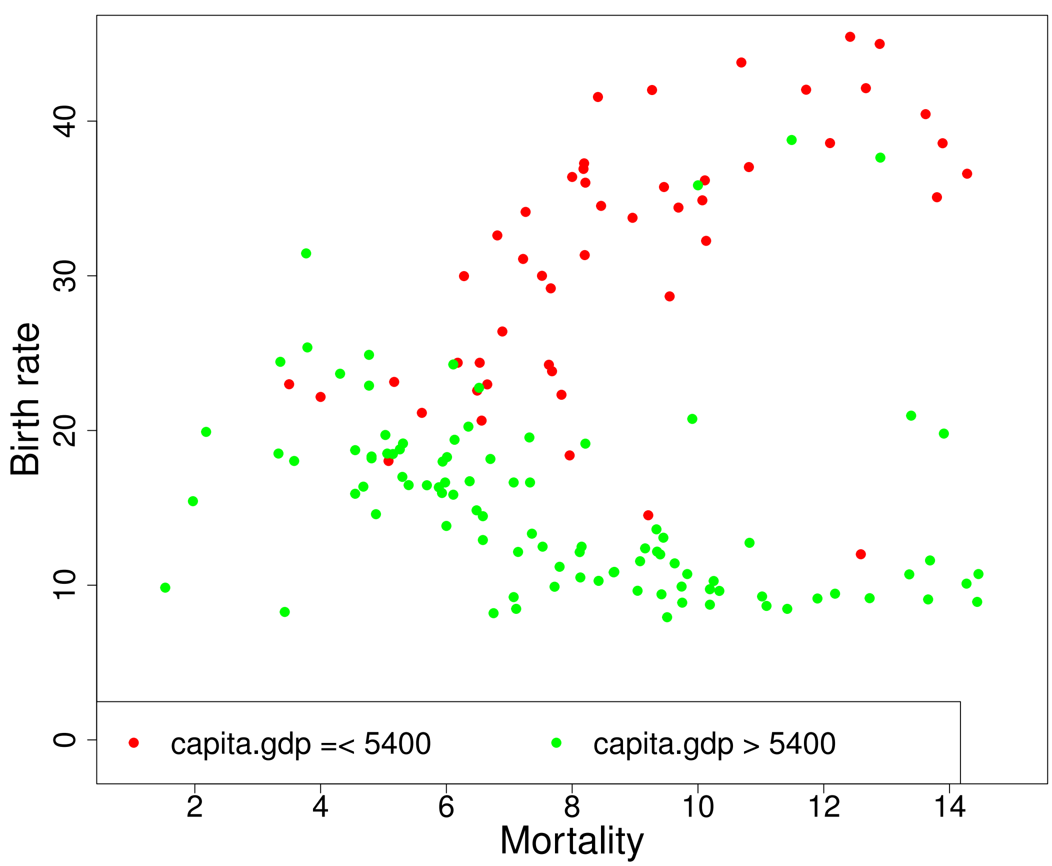

Fig. 1.2. Birth rate against mortality in 145 countries for two categories of GDP per capita

1.3.2. An example of pairwise independence

We wish to detect any form of dependence between variables, even 'weird' looking relations like in the previous example. But we also want to detect dependence between an arbitrary number of random variables (or vectors), not only two. This is because pairwise independence does not guarantee total mutual independence. Therefore, when dealing with more than two random vectors, a procedure testing only pairwise independence could fail to detect dependence when in fact there is, even if this procedure is consistent to detect any departure from independence.

To illustrate this, we present a second motivational example, taken from Romano and Siegel [28]. Let X, Y and Z_0 be independent random variables, of standard normal distribution. If we define Z as follows

$$Z = |Z_0| \cdot \text{sign}(X \cdot Y),$$

it yields that Z also has a standard normal distribution. Now, it can be shown that X, Y, Z are pairwise independent, but not mutually independent. On an intuitive level this means that information on X or Y does not bear any information on Z, and conversely. However, *joint* information on X and Y gives some information about Z. For instance:

$$\text{P}[Z > 0 | X > 0, Y > 0] = 1 \neq \text{P}[Z > 0] = 1/2.$$

Likewise:

$$\text{P}[X > 0, Y > 0, Z > 0] = 1/4 \neq \text{P}[X > 0] \cdot \text{P}[Y > 0] \cdot \text{P}[Z > 0] = 1/8.$$

To see graphically that the three variables X, Y, Z are pairwise independent we generate a sample from the joint distribution of (X, Y, Z). We display the pairwise scatter plots of the empirical cumulative distribution functions of the observations, $F_N(X), F_N(Y), F_N(Z)$, on Figures 1.3 to 1.5. As expected, they reveal no pattern. We plot the empirical CDF of the observations, rather than the observation themselves to facilitate visual examination. This choice will be explained with more details later on, in section 3.1.

For now, we have that all pairs are independent, and therefore one might be tempted to conclude that X, Y and Z are mutually independent. This wrong assumption might turn out to be quite harmful. For instance, once one assumes

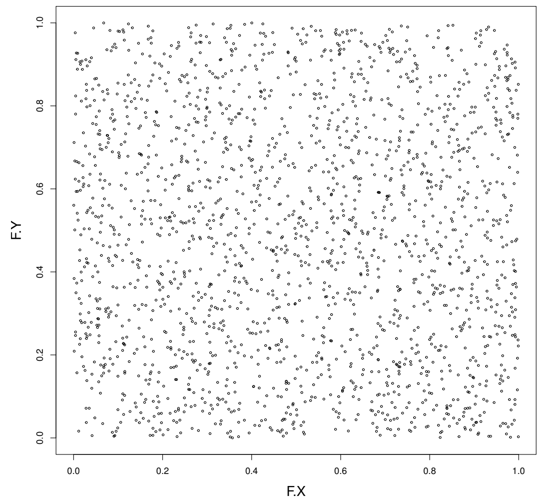

FIG. 1.3. Scatter plot of $F_N(Y)$ vs $F_N(X)$

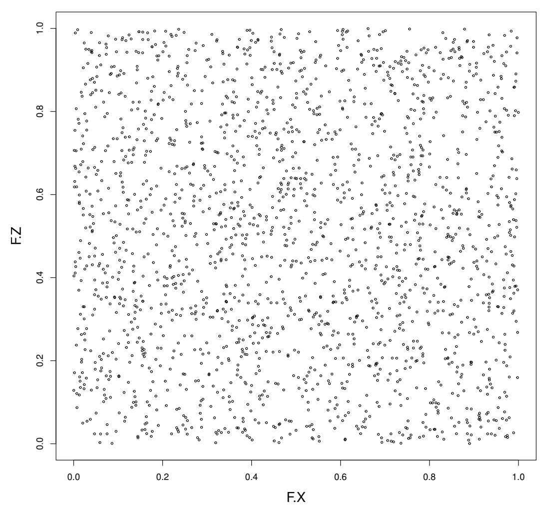

FIG. 1.4. Scatter plot of $F_N(Z)$ vs $F_N(X)$

mutual independence of X, Y and Z, one might furthermore conclude that:

$$S = X + Y + Z \sim N(0, \sqrt{3})$$

Which would be true if X, Y and Z where indeed mutually independent. However, in our present example, S is far from being $N(0, \sqrt{3})$. To illustrate this, we simulated a sample of $S = X + Y + Z$. Figure 1.6 shows the resulting (empirical) density of S, plotted against the density of a $N(0, \sqrt{3})$. They differ markedly.

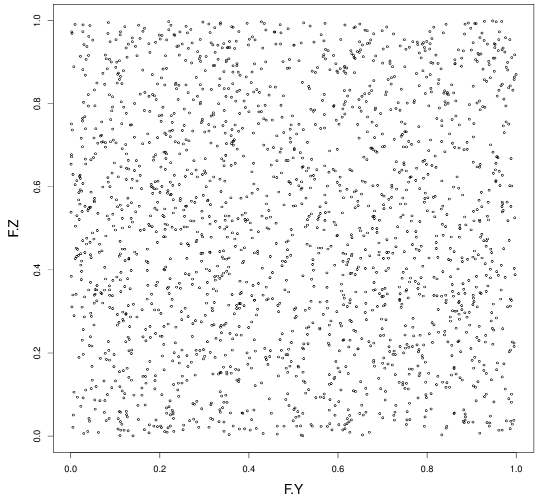

FIG. 1.5. Scatter plot of $F_N(Z)$ vs $F_N(Y)$

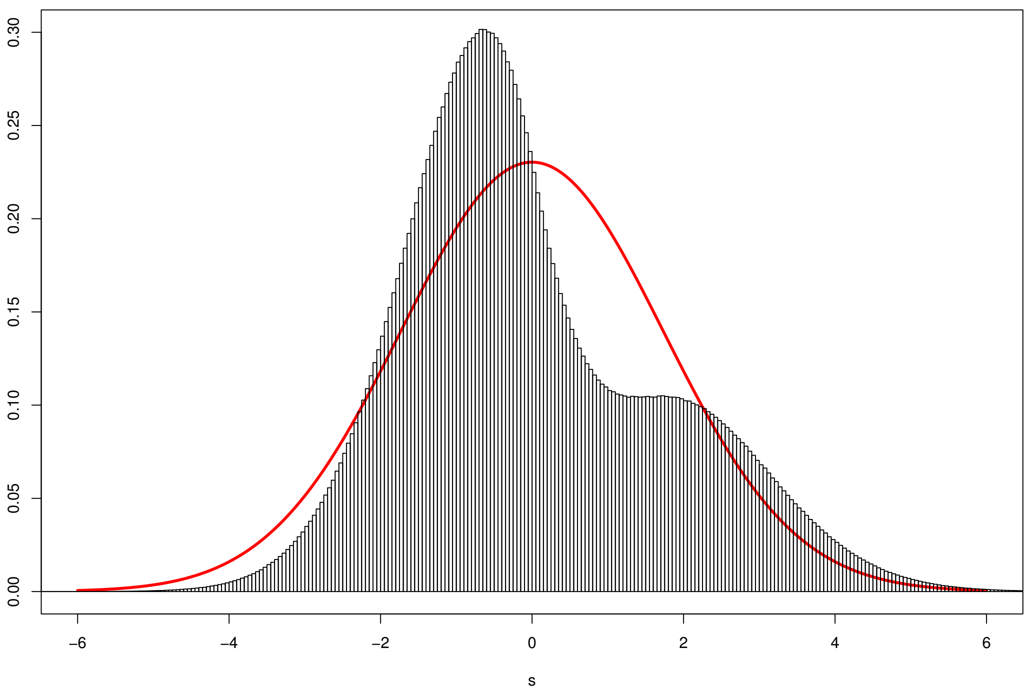

FIG. 1.6. Distribution of $X + Y + Z$ against that of a $N(0, \sqrt{3})$

To be fair, one could say that a 3D plot of $\{F_N(X), F_N(Y), F_N(Z)\}$ might have helped to reveal that $X, Y\ Z$ are not independent. Such a plot is displayed in Figure 1.7. Note that points with a positive value of Z are in blue and points with negative value of Z are in green. Examining this plot we realize that some areas contain no points at all. Having the model that generated the data in mind, this makes sense: if X and Y are both positive or both negative, then it is impossible for Z to be negative.

That being said, visual examination has great limitations. More subtle forms of dependence might be harder to see with the naked eye on a 3D plot. Moreover, past three dimensions, visualization becomes impossible. Also, in the presence of large data sets, it might not be feasible nor efficient to plot all pairs or triplets of variables to find associations between variables.

Hence, we need a systematic way to detect dependence of any form between an arbitrary number of random variables or vectors. This is precisely the scope of this research.

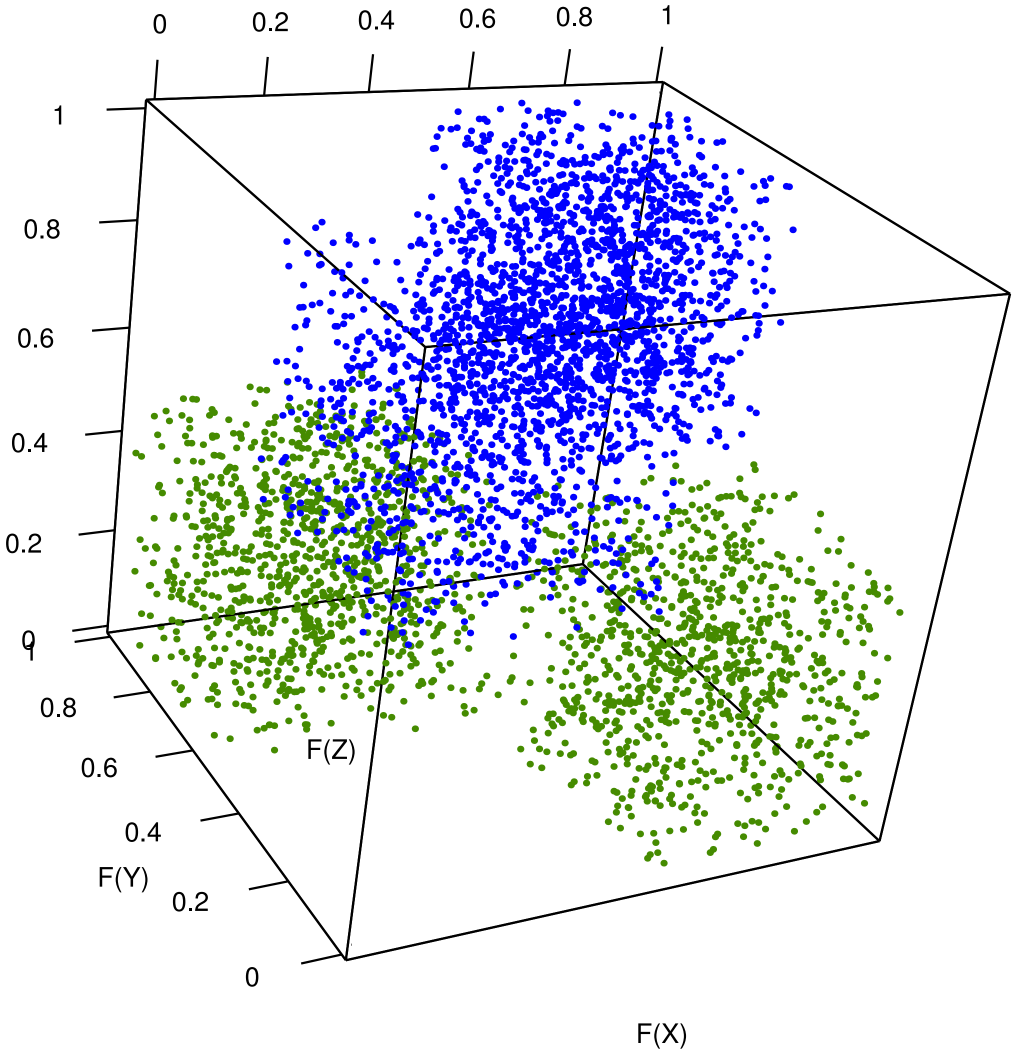

FIG. 1.7. Scatter plot of $F_N(X), F_N(Y), F_N(Z)$

1.4. Definitions of independence

To be a bit more formal, here we state some sufficient criteria for the independence of random variables. We consider first the case of two random variables, say X and Y.

1.4.1. Bivariate case

Probably the most common sufficient criteria to state the independence of two random variables X and Y is given in the following theorem.

Theorem 1.4.1. *X and Y are independent if and only if*

$$F_{XY}(x,y) = F_X(x)F_Y(y) \tag{1.4.1}$$

for all $x, y \in \mathbb{R}$.

A slightly different criteria, which will turn useful in the context of the present research is given next.

Theorem 1.4.2. *X and Y are independent if and only if*

$$P[x_1 \leq X \leq x_2; y_1 \leq Y \leq y_2] = P[x_1 \leq X \leq x_2] \cdot P[y_1 \leq Y \leq y_2] \tag{1.4.2}$$

for all $x_1, x_2, y_1, y_2 \in \mathbb{R}$.

Another alternative definition taken from Duan [7] is stated next.

Theorem 1.4.3. *X and Y are independent if and only if*

$$E[f(X)g(Y)] = E[f(X)]E[g(Y)] \tag{1.4.3}$$

for all Borel measurable and bounded functions f and g.

The same result holds if the restriction on f and g is that they are bounded and continuous, see Rényi [25].

1.4.2. Multivariate case

The two first definitions have 'obvious' generalizations in the multivariate case. We can get the first in any probability textbook, so for instance in Resnick [27], p. 94 we find:

Theorem 1.4.4. *A finite collection of random variables $X_1, \cdots, X_k$ is independent if and only if*

$$P[X_1 \le x_1, \cdots, X_k \le x_k] = \prod_{i=1}^{k} P[X_i \le x_i] \tag{1.4.4}$$

for all $x_i \in \mathbb{R}$.

Said otherwise, a collection of random variables are independent if and only if their joint distribution function is the product of the marginal distribution functions. Yet again, we can establish a slightly different result, which is handy in this research.

Theorem 1.4.5. *A finite collection of random variables $X_1, \cdots, X_k$ is independent if and only if*

$$P[a_1 \le X_1 \le b_1, \cdots, a_k \le X_k \le b_k] = \prod_{i=1}^{k} P[a_i \le X_i \le b_i] \tag{1.4.5}$$

for all $a_i, b_i \in \mathbb{R}$.

The proof of this result is given in Appendix A.1.

1.5. Outline

The present research is organized as follows. Chapter 2 reviews some important methods to detect dependence in the current literature. Starting with classical results, we move towards more recent, state-of-the art procedures, the last of those being Heller et al. [15] which is the starting point for our own test. In chapter 3 we propose a simple modification to the Heller et al. [15] procedure and we conduct power simulations to see if in the bivariate case this modification changes the performance of the test. Then in chapter 4 we extend their procedure to test for independence in a multivariate context. Again we present and discuss power simulations. Chapter 5 summarizes the relevant contributions we are making, presents some limits of the method as well as possible follow-up work.

Chapter 2

LITERATURE REVIEW

2.1. Overview

The problem of testing independence between variables is more than a hundred years old, and consequently the existing literature presents a variety of methods to address the question. We are interested in methods that apply in a general context. This means that the test must meet several requirements that we now explain, in relation to the existing literature.

First, we seek methods that do not make strong assumptions, for example on the distribution of the random variables in use. Classical rank methods (Spearman [33], Kendall [18] or Blomqvist [5]) satisfy this criterion. However, they have the important flaw that they can only detect certain forms of dependence.

Moreover, we want methods that allow to test the independence between more than two (ideally any) number of random vectors. Such methods exist, but often at the cost of distributional assumptions, for instance Wilks [40], extended in Wald and Brookner [38], assume multivariate normality, while Gieser and Randles [11], extended in Um and Randles [37], assume elliptical distributions.

The last, and arguably most difficult feature to obtain is that of universal consistency. It has been given a truly satisfying answer only with recent results such as Székely et al. [34], Gretton et al. [13] or Heller et al. [15], which however do not meet the second requirement as they are made to test independence between pairs of variables. Beran et al. [3] as well as the multivariate version of the Hilbert-Schmidt Independence Criterion (HSIC) found in Pfister et al. [23] meet both three of the above-mentioned requirements, and as such will be used as benchmarks for the test developed in the present research.

We now give details about some existing methods, starting with ‘classical’ results, and then moving on to recently proposed universally consistent procedures.

2.2. Classical dependence measures

2.2.1. Pearson's correlation

Pearson's correlation was first introduced by Galton [9], but its current formulation is due to Pearson [22]. Still vastly used today, this index is, as noted by Lee Rodgers and Nicewander [21], "remarkably unaffected by the passage of time". Hence, we start with a review of this crucial statistical tool.

Pearson's correlation is a standardization of the covariance. Let X and Y be two random variables with expectations $E[X] = \mu_x$, $E[Y] = \mu_y$, and variances $V[X] = \sigma_x^2$, $V[Y] = \sigma_y^2$ their correlation coefficient is given by

$$\rho_{X,Y} = \frac{\mathrm{COV}[X,Y]}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}. \tag{2.2.1}$$

Dividing by $\sigma_X \sigma_Y$ insures that the resulting index will be between -1 and 1. Roughly speaking, it will be positive if:

- When $X > \mu_X$, it is likely that $Y > \mu_Y$.
- When $X < \mu_X$, it is likely that $Y < \mu_Y$.

It takes the value 0 if and only if the covariance is 0. This requires the expectation of the product of X and Y to be equal to the product of the individual expectations,

$$E[XY] = E[X]E[Y]. \tag{2.2.2}$$

On the other hand, independence between the random variables X and Y means that the joint cumulative distribution function is equal to the product of the marginal distribution functions

$$F_{X,Y}(x,y) = F_X(x)F_X(x), \qquad \forall x, \forall y. \tag{2.2.3}$$

As already presented in section 1.4. Note that because (2.2.3) has to be true for every value x and y, independence is a much stronger assertion than non-correlation. In other words,

$$\text{independence of } X, Y \implies \rho_{X,Y} = 0. \tag{2.2.4}$$

But the implication in the other direction is false. Indeed, we can build several examples of dependent random variables that are uncorrelated. We can even have two deterministically related random variables (knowing one implies knowing the other), that still are not correlated, as shown in the following example.

Exemple 2.2.1. *Let X be a zero-mean random variable with a symmetric distribution, such as a $N(0,\sigma)$. Let $Y = X^2$, then*

$$E[XY] = E[X^3] = 0\ ;$$
$$E[X]E[Y] = E[X]E[X^2] = 0.$$

Following (2.2.2), *X and Y are not correlated, even though they are (strongly) dependent.*

This example reminds us that the correlation coefficient characterizes the *linear* trend existing between X and Y. This is one important but very specific type of relation. Correlation is unable to detect any other form of association. In the example, the relation between X and Y, although deterministic, was of quadratic form and hence correlation could not detect it.

To motivate this point further more, recall definition 1.4.3. For independence to hold, $E[f(X)g(Y)]$ must be equal to $E[f(X)]E[g(Y)]$ for *any* Borel measurable and bounded functions f and g. Uncorrelatedness is only one of such cases, that of $f(x) = x$ and $g(y) = y$, meaning it is a far weaker condition than independence.

Now, say we have collected a sample of (X, Y) and we want to use it to find the correlation between X and Y. Most of the time, we will not know from what distribution this data came from. Therefore, in practice we need the empirical version of Pearson's correlation, given by

$$r_{x,y} = \frac{\sum\limits_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y}. \tag{2.2.5}$$

Where $\bar{x}$ and s_x are respectively the mean and standard deviation of the collected sample from X, that is

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \qquad s_x^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2.$$

Note that what we mentioned earlier about the theoretical correlation still holds with its empirical counterpart: it will only detect a linear relationship between two variables.

One final comment to make is that correlation, even when it is significantly different than 0, might give limited information about the relation between X and Y. This is because in the end, it is just an index (one single number) and as such it cannot hold all the information contained in the sample. To illustrate this, let us use a famous example known as Anscombe's quartet, see Anscombe [1]. In this example, four data sets have the same significantly positive correlation of 0.816. However, they look vastly different, as shown in Figure 2.1.

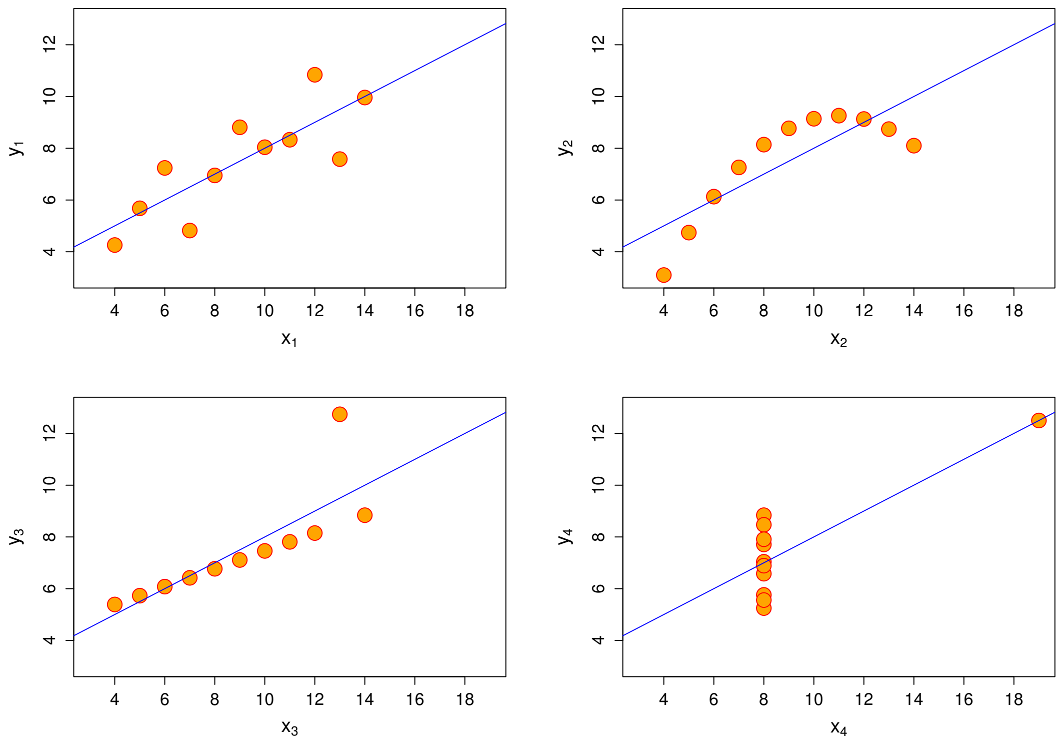

Fig. 2.1. Four (very) different sets of data with equal correlation

This means that although the correlation coefficient can be interpreted as the strength of the linear trend between two variables, it is far from delivering the whole picture.

2.2.2. Spearman's rho

Moving on, we present a few methods to detect dependence that are based on ranks. Provided that it is possible to compute ranks (which is not doable for instance with categorical data), these methods do not require any assumptions about the distribution of the random variables, and hence meet the first requirement listed in section 2.1. Furthermore, they bring something new compared to standard correlation because they detect not only linear but also monotone relations between variables. The first such method is due to Spearman [33]. We present the empirical version of what is now known as *Spearman's rho*.

Let $(x_1, y_1), \cdots, (x_n, y_n)$ be a sample of size n from some bivariate distribution (X, Y). Let R_i be the rank of observation x_i among the n observations from X. For instance, if x_i is the smallest X observed, then $R_i = 1$. Eventual ties between observations are dealt with using the average method (see Section 3.1 for details). Let S_i be the rank of observation y_i, defined in the same way. Spearman's rho, ρ_S is computed as Pearson's correlation coefficient of $(R_i, S_i), i \in (1, \cdots, n)$.

Alternatively, if we define a quantity D_i as follows:

$$D_i = R_i - S_i$$

then, if the n ranks (R_i, S_i) are all distinct integers, we have the compact formula:

$$\rho_S = 1 - \frac{6 \sum_{i=1}^{n} D_i^2}{n(n^2 - 1)}. \tag{2.2.6}$$

Spearman's rho is also an index between -1 et 1. Note that if $\sum_{i=1}^{n} D_i^2 = 0$, then $\rho_S = 1$. This happens when, within the sample of (X, Y), an observation x_i always has the same rank as its counterpart y_i. Stated otherwise, if an increase in X always corresponds to an increase in Y, then $\rho_S = 1$. Conversely, if the ranks of x_i are precisely the opposite of those of y_i, (an increase in X always corresponds to an decrease in Y) then $\rho_S = -1$.

Let us emphasize the fact that the range of relations detected by ρ_S is not those of strictly linear form. For instance, let us consider the data showed in Figure 2.2. Although the relationship between X and Y is not strictly linear (it is exponential), it is monotone and strictly increasing. Hence, in this case $\rho_S = 1$.

Consider the four sets of data of Anscombe's quartet on Figure 2.1. We compute the associated Spearman's rho, and corresponding p-values. From left to right and top to bottom the results are as follows

$$0.818 \ (p\text{-value} = 0.004)$$

$$0.691 \ (p\text{-value} = 0.023)$$

$$0.991 \ (p\text{-value} < 0.001)$$

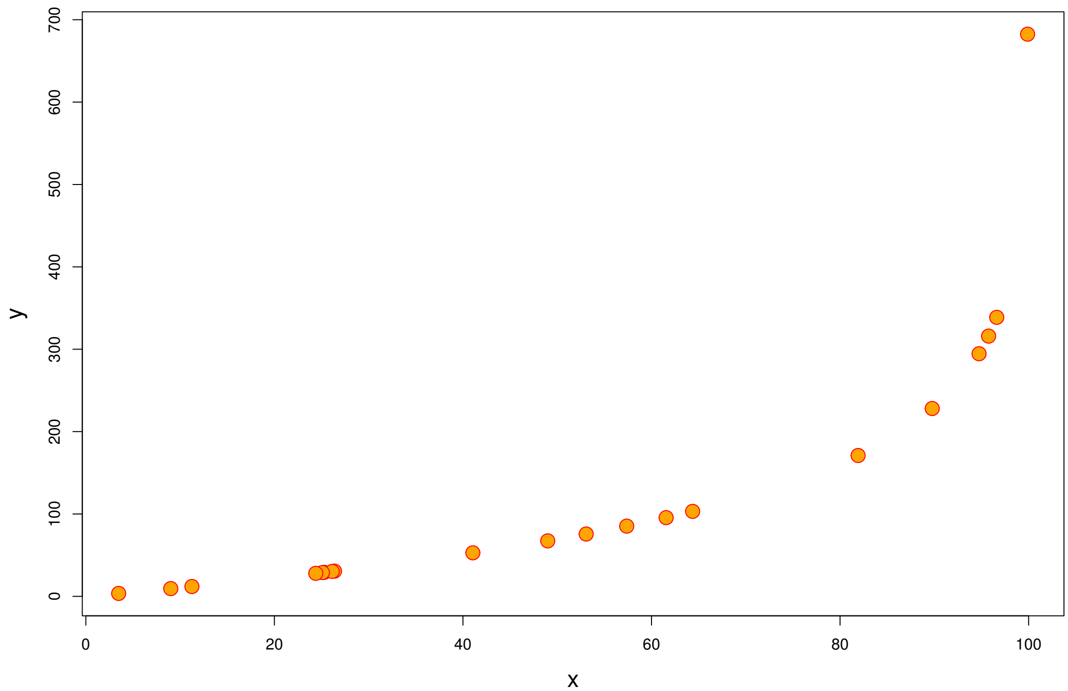

FIG. 2.2. A strictly monotone relation yields $\rho_S = \tau = 1$

$$0.500 \text{ (p-value} = 0.117).$$

Recall that in those examples Pearson's correlation was always the same (0.816), however the scores here differ markedly. Interestingly, for the last set of data the p-value is greater than 5% and we would not reject independence.

2.2.3. Kendall's tau

Another popular rank measure used to detected monotone forms of dependence between X and Y is due to Kendall [18]. Let (x_i, y_i) and (x_j, y_j) be two pairs of observations taken from a sample of size n from (X, Y). Such a pair is said to be *concordant* if $(x_i - x_j)(y_i - y_j) > 0$. It is said to be *discordant* if $(x_i - x_j)(y_i - y_j) < 0$. It is neither concordant nor discordant if $(x_i - x_j)(y_i - y_j) = 0$.

For n data points, there are in total $n(n-1)/2$ pairs of points. Let C be the number of concordant pairs and D be the number of discordant pairs. Then, Kendall's tau, τ, is given by:

$$\tau = \frac{2(C - D)}{n(n-1)}. \tag{2.2.7}$$

The index τ is between -1 and 1 and has an interpretation very similar to that of ρ_S: a value close to 1 means an increasing monotone relation between X and Y (such as shown in Figure 2.2), while a value close to -1 means a decreasing monotone relation. For completeness we again give the scores for Kendal's tau in the four datasets of Anscombe's quartet

$$0.636 \ (p\text{-value} = 0.006)$$
$$0.564 \ (p\text{-value} = 0.017)$$
$$0.964 \ (p\text{-value} < 0.001)$$
$$0.426 \ (p\text{-value} = 0.114).$$

Note that a relatively recent paper by Taskinen et al. [36] develops generalizations of both Spearman's rho and Kendall's tau. This paper provides two new statistics that allow to test the independence of two random *vectors* X and Y in arbitrary, possibly different dimensions. These statistics have a convenient limiting χ^2 distribution and are robust to outliers.

2.2.4. Wilks and Puri-Sen statistics

The methods presented so far are not only limited because they detect few forms of dependence, but also because they are *pairwise* indexes. If we are interested in detecting dependence between several variables, we can start with a paper due to Wilks [40].

In this paper, a statistic λ is derived as a Neyman-Pearson ratio using the maximum likelihood principle. Hence this test is often called the likelihood ratio test (LRT) of independence. It is meant to test the mutual independence of a set of k sets of random variables with a multivariate normal distribution. Hence, contrary to the methods presented before, we can now test the independence between more than two random variables. Furthermore, those random variables can be *sets*, meaning they can themselves be in several dimensions.

Note that for the problem of testing the independence between $X \in \mathbb{R}^p$ and $Y \in \mathbb{R}^q$, two vectors with multivariate normal distributions, the test statistic W can be written as:

$$W = 2 \log \lambda = -n \log \det(I - S_{22}^{-1} S_{21} S_{11}^{-1} S_{12}), \tag{2.2.8}$$

where S_{11} is the sample covariance matrix of X, S_{22} is the sample covariance matrix of Y, S_{12} is the sample covariance matrix of (X, Y), and S_{21} is the sample covariance of (Y, X), which is equal to S_{12}^T. Under multivariate normality of X and Y the distribution of W is known as the Wilks Lambda distribution $\Lambda(q, n-1-p, p)$.

Compared to correlation or rank coefficients, this method can now answer a more *complex* question, because we can test the independence of more than two variables. Moreover, those variables, $X, Y, Z, \cdots$ are also allowed to be vectors. However, we have lost some generality because we have to assume multivariate normality of our observations. As noted in Puri and Sen [24], for non-normal distributions the correlation matrix might not exist, or if it exists it "may not play the fundamental role that it does in the case of the multinormal distributions" (because then uncorrelation does not imply independence), and hence there is a need for a less restrictive approach. The class of tests then described in chapter 8 of Puri and Sen [24] are said to be analogues to the original Wilks test (Bakirov et al. [2]). However, they rely on other sample dispersion matrices rather than covariance or correlation matrices. Say that $T = (T_{ij})$ is such a sample dispersion matrix. Then, the test simply replaces the covariances matrices $S_{11}, S_{22}, S_{12}, S_{22}$ in 2.2.8 by their analogues in T. For instance, T could be taken as the matrix of Spearman's rho statistics.

2.2.5. Hoeffding's D

Another classical result, free of any assumption about the distribution of the random variables, is found in Hoeffding [16]. Wilding and Mudholkar [39] state that although it is an important result, because it is not straight-forward to use it is "largely ignored in application".

The idea behind the test is intuitive: let $D(x, y) = F_{X,Y}(x, y) - F_X(x)F_Y(y)$. Then, $D(x, y) = 0 \ \forall (x, y)$ if and only if X and Y are independent. Again denote by R_i and S_i the respective ranks of x_i and y_i in a collected sample of size n. Also, define c_i as the number of bivariate observations (x_j, y_j) for which $x_j \leq x_i$ and $y_j \leq y_i$. Then, the quantity $\int D^2(x, y)\mathrm{d}F(x, y)$ has the non-parametric estimator

$$D_n = \frac{Q - 2(n-2)R + (n-2)(n-3)S}{n(n-1)(n-2)(n-3)(n-4)}.$$

Where

$$Q = \sum_{i=1}^{n} (R_i - 1)(R_i - 2)(S_i - 1)(S_i - 2),$$

$$R = \sum_{i=1}^{n} (R_i - 2)(S_i - 2)c_i,$$

$$S = \sum_{i=1}^{n} (c_i - 1)c_i.$$

The difficulty of using D_n in practice is that its distribution is unknown, and according to Wilding and Mudholkar [39], "lack of distributional approximations makes it difficult to obtain p-values except by rough interpolation." The authors of this paper then present a method to approximate the null distribution of D_n.

Note at this point we are still in a quest for universal consistency, because, as stated in Kallenberg and Ledwina [17], "Hoeffding's test may completely break down for alternatives that are dependent but have low grade linear correlation."

2.3. Recent developments

The results presented before are somehow 'classical', and although still vastly used they present important limitations, as they cannot detect some forms of dependence, or because they make distributional assumptions. We now present some more recent results that have the desirable property of being consistent against every dependence alternative, or at least a wide variety of dependence alternatives.

2.3.1. Kallenberg and Ledwina [17]

Kallenberg and Ledwina [17] develop a test of independence between two continuous random variables X and Y. Their test is consistent against a "broad class" of alternatives, although not all alternatives. Denoting $X^* = F(X)$ and $Y^* = F(Y)$, they consider h, the joint cdf of X^* and Y^*. They call h the grade representation of X and Y, although a more common name would be 'copula'. Then their procedure tests $H_0 : h(x^*, y^*) = 1$, which corresponds to independence, against H_1 defined as

$$H_1 : \quad h(x^*, y^*) = c(\boldsymbol{\theta})\exp\left\{\sum_{j=1}^{k} \theta_j b_j(x^*) b_j(y^*)\right\}.$$

This is the exponential family for the joint distribution of X^* and Y^*, where the θ_j are constants and the b_j are the Fourier coefficients:

$$\begin{aligned} b_1(x) &= \sqrt{3}(2x - 1), \\ b_2(x) &= \sqrt{5}(6x^2 - 6x + 11), \\ &\text{etc.} \end{aligned}$$

Intuitively, we can say that this method tests if there is correlation between order polynomials (up to some order k) of X^* and Y^*. This is far more general than for instance Spearman's ρ, which tests the first order correlation of X^* and Y^*. The test statistic uses ranks of the observations, say $R_i = \text{rank}(x_i)$ and $S_i = \text{rank}(y_i)$ from a sample of size N, then

$$T_k = \sum_{j=1}^{k} \left\{ \frac{1}{\sqrt{N}} \sum_{i=1}^{N} b_j \left(\frac{R_i - \frac{1}{2}}{N} \right) b_j \left(\frac{S_i - \frac{1}{2}}{N} \right) \right\}$$

where the selection of the order k in T_k is done *à la* Schwarz's rule.

2.3.2. Genest and Rémillard [10]

Most tests presented up until now were concerned in detecting dependence between *two* random variables. Genest and Rémillard [10] propose a test of independence between an *arbitrary* number of continuous random variables, say $X_1, ..., X_p$. They argue it is 'widely recognized' that the dependence structure of a set of random variables is best characterized by their copula function C (as opposed to their joint cdf F). Recall the copula function is nothing else then the cdf of the vector $(U_1, \cdots U_p)$ where $U_j = F_j(X_j)$. Hence they develop a test based on the empirical copula C_n, which is constructed from a collected sample of size n from $X_1, ..., X_p$, that is

$$C_n(u_1, \cdots, u_p) = \frac{1}{n} \sum_{i=1}^{n} \prod_{j=1}^{p} \mathbb{1} \left\{ \frac{R_{ij}}{n+1} \leq u_j \right\}.$$

where R_{ij} is the rank of the ith observation for variable j among the total n observations, that is

$$R_{ij} = \sum_{l=1}^{n} \mathbb{1} \left\{ X_{lj} \leq X_{ij} \right\}, \quad 1 \leq i \leq n, \quad 1 \leq j \leq p.$$

Another quantity of interest is the copula process, which is the difference between the empirical and theoretical copulas, times the square of the sample size n:

$$\mathcal{C}_n(u_1, \cdots, u_p) = \sqrt{n} \{ C_n(u_1, \cdots, u_p) - C(u_1, \cdots, u_p) \}.$$

Now, Deheuvels [6] decomposed $\mathcal{C}_n$ into sub-processes $\mathcal{G}_{A,n}$ whose index A indicates a subset of $\{(1, \cdots p)\}$ with $|A| > 1$. Genest and Rémillard [10] exploit the fact that the $\mathcal{G}_{A,n}$ processes are asymptotically independent and Gaussian (under H_0) to construct a formal test of independence, as well as a randomness test (white noise test) in a serial context. One advantage of their approach is that it yields test statistics of simple form, whose computation only requires the ranks of the observations, for each of the p variables. In the non-serial setting (which interests us in the course of this research), we get the following series of test statistics $T_{A,n}$

$$T_{A,n} = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{n} \prod_{j \in A}^{n} \left\{ \frac{2n+1}{6n} + \frac{R_{ij}(R_{ij}-1)}{2n(n+1)} + \frac{R_{kj}(R_{kj}-1)}{2n(n+1)} - \frac{\max(R_{ij}, R_{kj})}{n+1} \right\}.$$

Each of these statistics can be used to conduct 'individual' tests of independence, to test the independence of any particular subset A of the p variables. Furthermore, by combining the p-values of those statistics (there are $2^p - p - 1$ of them), a test of mutual (i.e. total) independence can be performed. An extensive series of simulations were run, and the test turned out to be consistent for a variety of different alternatives. The power was also compared to that of the 'classic' likelihood ratio test (LRT) due to Wilks [40]. The simulations yielded a better or similar power for the new test when compared to the LRT, except in the case of multivariate normality. This was to be expected as the LRT is known to be optimal in that case.

2.3.3. Distance covariance

An important result in recent statistics is the concept of *distance covariance.* This procedure has become increasingly popular and generated a lot of follow-up work since its first appearance. It takes its origins in Bakirov et al. [2], but was established in its definitive form in Székely et al. [34] and further extended and justified in Székely et al. [35].

Distance covariance, noted $\mathcal{V}^2(X,Y)$ is a measure of the distance between the joint characteristic function of $X \in \mathbb{R}^p$ and $Y \in \mathbb{R}^q$ and the product of their marginal characteristic functions:

$$\mathcal{V}^2(X,Y) = ||f_{X,Y}(t,s) - f_X(t)f_Y(s)||_w$$

where $||\cdot||_w$ is a norm (in the form of an integral), defined for a carefully chosen weight function w. That is:

$$\mathcal{V}^2(X,Y) = \int_{\mathbb{R}^{p+q}} |f_{X,Y}(t,s) - f_X(t)f_Y(s)|^2 w(t,s)dt\ ds.$$

In practice, the empirical version of $\mathcal{V}^2(X,Y)$ is used, noted $\mathcal{V}_n^2(X,Y)$. It is shown that

$$\lim_{n\to\infty} \mathcal{V}_n^2(X,Y) = \mathcal{V}^2(X,Y).$$

To define $\mathcal{V}_n^2(X,Y)$, let us again consider a sample of size n collected from $(X \in \mathbb{R}^p, Y \in \mathbb{R}^q)$, $\{(x_i,y_i) : k = 1, \cdots, n\}$ and define

$$a_{kl} = |x_k - x_l|_p, \quad \bar{a}_{k\cdot} = \frac{1}{n}\sum_{l=1}^{n} a_{kl}, \quad \bar{a}_{\cdot l} = \frac{1}{n}\sum_{k=1}^{n} a_{kl},$$

$$\bar{a}_{\cdot\cdot} = \frac{1}{n^2}\sum_{k,l=1}^{n} a_{kl}, \quad A_{kl} = a_{kl} - \bar{a}_{k\cdot} - \bar{a}_{\cdot l} + \bar{a}_{\cdot\cdot}$$

where $|\cdot|$ is the euclidean norm. Define analogue quantities for Y: $b_{kl} = |y_k - y_l|_p$, etc. Then, the empirical distance covariance is defined by

$$\mathcal{V}_n^2(X,Y) = \frac{1}{n^2}\sum_{k,l=1}^{n} A_{kl}B_{kl}.$$

Similarly, the distance variance $\mathcal{V}_n^2(X)$ is defined as

$$\mathcal{V}_n^2(X) = \mathcal{V}_n^2(X,X) = \frac{1}{n^2}\sum_{k,l=1}^{n} A_{kl}^2.$$

Finally, the empirical distance-correlation is then

$$\mathcal{R}_n = \frac{\mathcal{V}_n^2(X,Y)}{\sqrt{\mathcal{V}_n^2(X)\mathcal{V}_n^2(Y)}}.$$

Probably the most important feature of distance-correlation is that $\mathcal{R}(X,Y) = 0$ if and only if X and Y are independent. As such, it means that distance-correlation *characterizes* independence and is therefore consistent in every situation. Tests of independence based on distance-covariance most commonly are conducted using a permutation procedure. As the test we are building also relies on such a procedure, this will be explained in details later.

Note that in the above-mentioned papers, such a test of independence is compared empirically with the classical test in Wilks [40], and with two versions of the tests by Puri and Sen [24] for multivariate independence. It is found that the new test has superior or similar power then the other tests (depending on the situation), and that the proposed statistic is sensitive to all types of departures from independence (nonlinear, nonmonotone, etc.).

2.3.4. BBL test

Another existing test that is universally consistent and applicable to several random vectors is found in Beran et al. [3]. We refer to it as the BBL test. It uses an idea similar to that of Székely et al. [34] in that it measures the distance between the (empirical) distribution functions of the random vectors. However it uses a different weight function. Not only is this method applicable to detect dependence between any number of random vectors, but it simultaneously tests which subsets of vectors are independent. It also provides a visual tool called *dependogram* to visualize which subsets of vectors are dependent.

No distribution assumptions are made about the random vectors, and as such this method can be seen as a follow up to a paper previously published by Bilodeau and Lafaye de Micheaux [4] that assumed normal margins.

Because this test is consistent against every alternative, and implementation is available in a R package, it will serve as a benchmark for the test we propose in the current research.

2.3.5. Hilbert-Schmidt Independence Criterion

In Gretton et al. [13] and Gretton et al. [14] a universally consistent test of independence is presented, which we will refer to as the HSIC test. They develop a theoretical measure of dependence between two random vectors X and Y, which can be written as:

$$\mathrm{HSIC}(p_{xy}, \mathcal{F}, \mathcal{G}) := ||C_{xy}||_{HS}^2$$

where

- $\mathcal{F}$ and $\mathcal{G}$ denote sets of functions (called Reproducing Kernel Hilbert Spaces) containing all continuous bounded real-valued functions of X and Y, respectively.
- $C_{xy} : \mathcal{G} \to \mathcal{F}$ is called the cross-covariance operator, and is the unique operator mapping elements of $\mathcal{G}$ to elements of $\mathcal{F}$ such that for $\langle \cdot, \cdot \rangle$ an inner-product:

  $$\langle f, C_{xy}(g) \rangle_{\mathcal{F}} = \mathrm{COV}(f, g)$$

  for all $f \in \mathcal{F}$ and all $g \in \mathcal{G}$.
- $|| \cdot ||_{HS}^2$ is called the Hilbert-Schmidt norm of an operator.

Importantly, if this theoretical measure $||C_{xy}||_{HS}^2$ is zero, then it means $\langle f, C_{xy}(g) \rangle_{\mathcal{F}}$ and hence $\mathrm{COV}(f, g)$ is zero for any $f \in \mathcal{F}$ and any $g \in \mathcal{G}$. But remembering theorem 1.4.3 this ensures that X and Y are independent. $||C_{xy}||_{HS}^2$ being zero is then a sufficient condition for independence.

Of course we need an empirical estimate (i.e. a test-statistic) of this theoretical measure. Such a quantity is developed in Gretton et al. [13], and is proven to converge to the theoretical measure when the sample size increases. Then Gretton et al. [14] state that the distribution of this test statistic is ‘complex’, and hence they propose to use a permutation method to calculate its p-value. They also propose an approximation using the first two moments of the test statistic and a Gamma distribution. This method is computationally way faster then the permutation method.

Note there has been a recent generalization of HSIC, which we label mHSIC, testing the independence between an arbitrary number of random vectors, see Pfister et al. [23]. We will also use this generalization to compare the power of our own test.

2.3.6. Maximal information coefficient

A recent development, although applicable in a less general framework because valid only for univariate X and Y is found in Reshef et al. [26]. As stated by the authors, the intuitive idea behind their test is that "if a relationship exists between two variables, then a grid can be drawn on the scatterplot of the two variables that partitions the data to encapsulate that relationship".

Hence, the procedure aims at finding the grid that displays the stronger relation. For a specific a-by-b grid, the criterion used to judge the strength of the relationship is the mutual information $I(\cdot, \cdot)$

$$I(X,Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right).$$

Where $p(x,y), p(x), p(y)$ are the empirical joint and marginal probability distribution functions induced by the a-by-b grid. All grids, up to a maximal resolution which is determined by the size of the sample are explored. For a given grid defined by the pair of integer (a, b) the maximal mutual information achieved is normalized, to ensure a fair comparison between grids of different resolutions. Call this normalized score $m_{a,b}$. MIC corresponds to the highest score in the matrix containing all values of $m_{a,b}$.

The authors promote their method not only stating that it can capture a "wide range of interesting associations", but also stating that their test has the property of *equatability*, in that it gives "similar scores to equally noisy relationships of different types. " However, this last statement has been contradicted by some, including Gorfine et al. [12]. In their comment, they also expose the fact that for what they call *practical sample sizes* (30, 50, 100), as opposed to the larger samples sizes used in their paper (250, 500, 1000), the HHG test in Heller et al. [15] as well as the distance correlation "hold very large power advantages over the MIC test". Another critique comes from Simon and Tibshirani [32]. These authors wrote that the set of simulations they conducted "suggests that MIC has serious power deficiencies, and hence when it is used for large-scale exploratory analysis it will produce too many false positives".

2.3.7. HHG test

Heller et al. [15] propose a remarkably simple test of independence, which we will refer to as the HHG test from now on. It has been found by the authors that their method is more powerful (sometimes drastically) than distance-covariance (dCov) in a number of examples. They conclude saying "we expect that the new test will perform better than dCov when the linear component in the relationship between X and Y is weak or entirely absent, as well as when the first moments of X and Y are large or infinite".

As our own methodology is an extension of theirs, we now conduct a detailed review of their method. Time spent doing this is not wasted, as it will make the generalized case we present in section 4.2 more intelligible. First, in section 2.3.7.1 we explain the approach in an intuitive manner, and then in section 2.3.7.2 the test is presented more rigorously.

2.3.7.1. *General idea*

Consider two random vectors $X \in \mathbb{R}^p$ and $Y \in \mathbb{R}^q$, where p and q are any integers. The goal is to test if there is any association between them. Specifically, we want to test the null hypothesis:

$$H_0 : F_{XY}(x, y) = F_X(x)F_Y(y) \quad \forall x \in \mathbb{R}^p, y \in \mathbb{R}^q.$$

To motivate intuitively this test, we start with an example where X and Y are both random variables ($p = q = 1$). In that case, it is useful to have in mind the alternative, equivalent definition of H_0 given by theorem 1.4.2:

$$H_0 : \mathrm{P}[X \in A, Y \in B] = \mathrm{P}[X \in A] \cdot \mathrm{P}[Y \in B],$$

$$\begin{aligned} &\forall A = [x_1, x_2] \quad \text{with } x_1, x_2 \in \mathbb{R} \text{ and } x_1 \le x_2 \\ &\forall B = [y_1, y_2] \quad \text{with } y_1, y_2 \in \mathbb{R} \text{ and } y_1 \le y_2. \end{aligned}$$

We will now explain the test using a real-data example which was introduced in section 1.3.1. Let X and Y be the 2015 overall mortality and birth rates, respectively, from a sample of 145 countries having a population larger than 2 millions.

This example serves our purpose because, as noted before, the scatter plot of observations reveals a pattern between mortality and birth rate, however the correlation coefficient is low (0.139) and not significantly different than zero (p-value $= 0.095$). The same conclusion is drawn if we compute Spearman's (-0.035, p-value $= 0.675$) or Kendall's (-0.032, p-value $= 0.574$) coefficients instead.

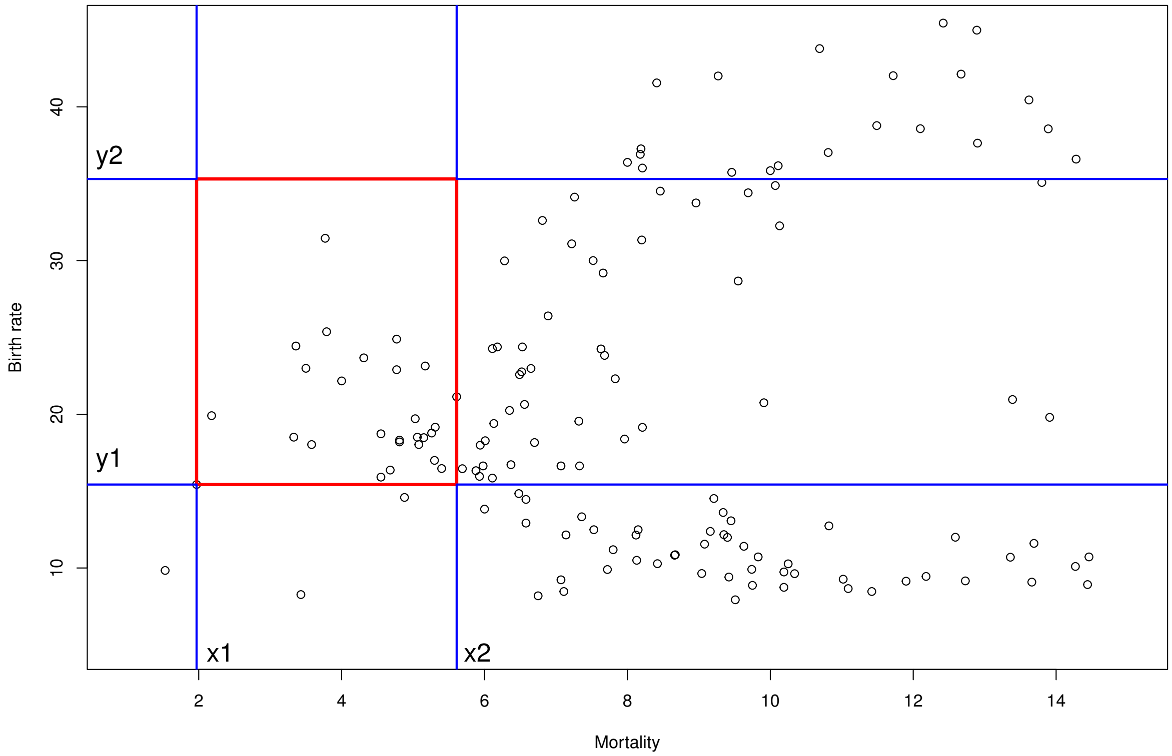

FIG. 2.3. 2015 birth rate against mortality rate in 145 countries

On Figure 2.3 we display this data, but we have placed a red rectangle of arbitrary size in an arbitrary location on the scatter plot. This rectangle has vertices (x_1, y_1), (x_2, y_1), (x_1, y_2) and (x_2, y_2), and we use it to motivate the test.

Let $A = [x_1, x_2]$ and $B = [y_1, y_2]$. If H_0 is true, then all the following equalities should be true:

$$\begin{aligned} \mathrm{P}[X \in A, Y \in B] &= \mathrm{P}[X \in A] \cdot \mathrm{P}[Y \in B] \\ \mathrm{P}[X \in A, Y \in B^C] &= \mathrm{P}[X \in A] \cdot \mathrm{P}[Y \in B^C] \\ \mathrm{P}[X \in A^C, Y \in B] &= \mathrm{P}[X \in A^C] \cdot \mathrm{P}[Y \in B] \\ \mathrm{P}[X \in A^C, Y \in B^C] &= \mathrm{P}[X \in A^C] \cdot \mathrm{P}[Y \in B^C]. \end{aligned} \tag{2.3.1}$$

With a collected sample of size N, we can estimate these probabilities by their empirical counterparts, for instance:

$$\mathrm{P}[X \in A, Y \in B] \approx \frac{\#\text{ points} \in A \text{ and} \in B}{N}$$

and:

$$\mathrm{P}[X \in A] \cdot \mathrm{P}[Y \in B] \approx \frac{\#\text{ points} \in A}{N} \cdot \frac{\#\text{ points} \in B}{N}.$$

Hence, to establish that H_0 is false, it would suffice to establish that any of the empirical versions of the probabilities on the left-side of equations (2.3.1) differ significantly from the empirical probabilities on the right-side of these equations.

A way to test *at the same time* the four equalities in (2.3.1) is to use the Pearson χ^2 test with the null hypothesis:

$$H_0 : \text{“being in A” is independent of “being in B”}$$

This yields a 2×2 contingency table. Under H_0 we can compute the expected counts in each of the four cells of this table, and then use the χ^2 statistic to test if they are significantly different from the observed counts. For the specific choice of rectangle we displayed on Figure 2.3, tables of observed and expected counts are given below. We also present an example on how to calculate the expected counts.

TAB. 2.1. Observed counts O_i

	$y \in B$	$y \notin B$
$x \in A$	27	2
$x \notin A$	44	72

TAB. 2.2. Expected counts E_i

	$y \in B$	$y \notin B$
$x \in A$	14.2	14.8
$x \notin A$	56.8	59.2

$$\begin{aligned} E_{11} =& N \cdot \mathrm{P}[\text{being inside in "x"}] \cdot \mathrm{P}[\text{being inside in "y"}] \\ =& N \cdot \frac{\#\text{ points inside in "x"}}{N} \cdot \frac{\#\text{ points inside in "y"}}{N} \\ =& \frac{29 \cdot 71}{145} = 14.2 \end{aligned}$$

Then using those contingency tables we compute the χ^2 statistic that tests the independence between categories:

$$\chi^2 = \sum_{i}^{4} \frac{[O_i - E_i]^2}{E_i} = 28.26 > \chi^2_{1,0.05}$$

In this specific example (with this specific choice of rectangle), we would then reject H_0 (quite strongly).

However, it is important to note that another choice of rectangle could have yielded a different conclusion. Furthermore, not being able to reject H_0 with a specific choice of rectangle does not mean H_0 is true. Then, the natural subsequent question is: how do we fix the location and size of the rectangle ? In the words of Heller et al. [15], "we let the data guide us."

Let us consider an arbitrary pair of data points $\{(x_i, y_i), (x_j, y_j)\}$ with $i \neq j$. Perhaps in a slight abuse of notation we just call this pair (i, j) from now on. We use this pair of points (i, j) to define a specific rectangle, as show in Figure 2.4. The point i defines the center of the rectangle, while the relative distance between i and j is used to define the size of the rectangle. For this particular choice, we categorize all the remaining $N-2$ points (those that are not i nor j) in the 2×2 contingency table presented earlier and compute the associated χ^2 test statistic. Let us call this statistic $S(i, j)$. The core of the method is to do this for all the possible pairs of points in the sample. Then, we take as our overall test statistic the sum of all those χ^2 statistics $S(i, j)$:

$$T = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N} S(i, j)$$

Note that there are $N(N-1)$ such pairs of points. In the words of Heller et al. [15], doing this "aggregates the evidence against independence". Note that if T is "big" (compared to what it would be under H_0), then it means that for at least one rectangle the test statistic $S(i, j)$ is "big" (compared to what it would be under H_0), therefore we have evidence that H_0 is false. Hence, we reject H_0 for big values of T. Because the distribution of T is unknown, a permutation method is required to compute its p-value, see section 3.2 for details.

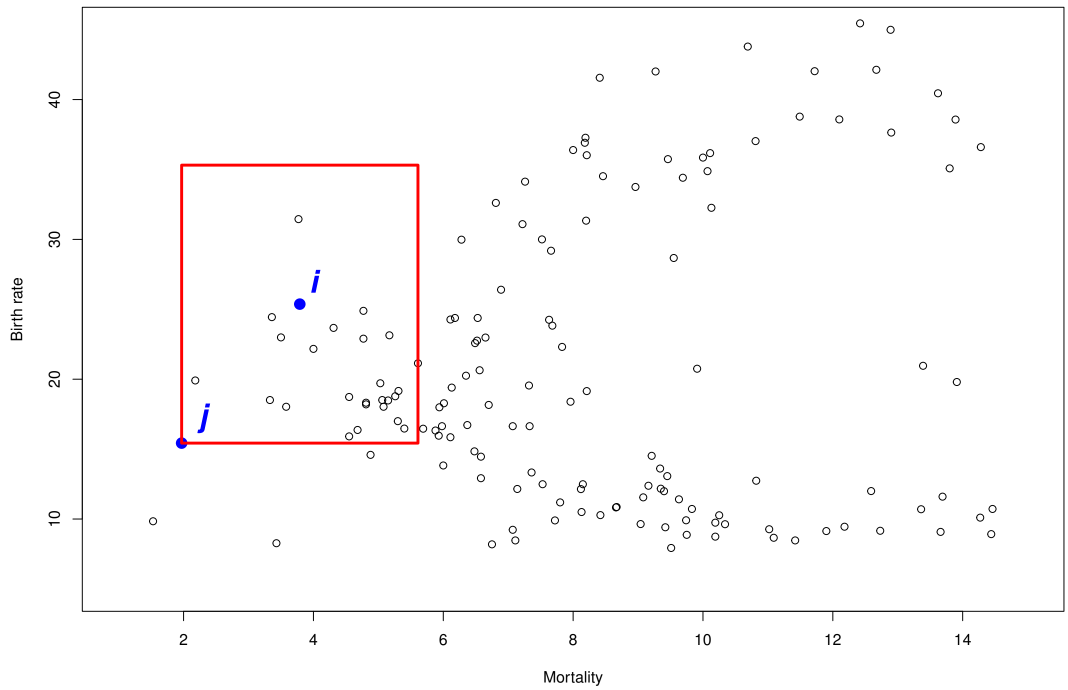

FIG. 2.4. The red rectangle is defined by the two blue points i and j

2.3.7.2. *Test of independence*

Most of the work is done, but we now present the test in a more formal way, using the notation in Heller et al. [15]. First note that if X and Y are vectors, we no longer have zones defined by rectangles. However, the general idea of the test remains the same.

Again, consider two arbitrary samples points (x_i, y_i) and (x_j, y_j) from the joint distribution of (X, Y). Then define two radii R_{x_0} and R_{y_0} as the euclidean distances $\text{dist}(\cdot, \cdot)$ between these points, according respectively to their X and Y coordinates[1]:

$$R_{x_0} = \text{dist}(x_i, x_j), \quad R_{y_0} = \text{dist}(y_i, y_j).$$

R_{x_0} and R_{y_0} are called radii because we use them to define a zone, centered at (x_i, y_i), in which the other points can be located or not. Consider for instance some sample point (x_k, y_k), with $k \neq i$ and $k \neq j$. According to the X coordinate,

[1]The article mentions that any other norm could be used instead. In fact, we could even use different norms for the distances between $(x_i\ , x_j)$ and $(y_i\ , y_j)$. However, in the present research we will stick to euclidean distances.

we categorize this point as being as close or closer to i then j is (hence within the radii R_{x_0}), or further form i then j is (outside the radii). We do the same according to the Y coordinate. Then, for the choice of i and j we made, we define the indicator functions:

$$\mathbb{1}\{\mathrm{dist}(x_i, x_k) \leq R_{x_0}\}$$
$$\mathbb{1}\{\mathrm{dist}(y_i, y_k) \leq R_{y_0}\}.$$

Again we have four categories, and we count the number of observations falling into each of these categories. For instance, the number of points *as close or closer to i then j is, according to both coordinates X and Y* is given by:

$$A_{11}(i,j) = \sum_{k=1, k\neq i, k\neq j}^{N} \mathbb{1}\{\mathrm{dist}(x_i, x_k) \leq R_{x_0}\} \cdot \mathbb{1}\{\mathrm{dist}(y_i, y_k) \leq R_{y_0}\}.$$

The quantities $A_{12}(i,j), A_{21}(i,j), A_{22}(i,j)$ are defined similarly. The cross-classification of the $N-2$ points (not equal to i or j) is summarized in a table of the following form:

TAB. 2.3. Categorization of $\mathbb{1}\{\mathrm{dist}(x_i, X) \leq \mathrm{dist}(x_i, x_j)\}$ and $\mathbb{1}\{\mathrm{dist}(y_i, Y) \leq \mathrm{dist}(y_i, y_j)\ \}$

	$d(y_i, \cdot) \leq \mathrm{dist}(y_i, y_j)$	$\mathrm{dist}(y_i, \cdot) > d(y_i, y_j)$
$\mathrm{dist}(x_i, \cdot) \leq \mathrm{dist}(x_i, x_j)$	$A_{11}(i,j)$	$A_{12}(i,j)$
$\mathrm{dist}(x_i, \cdot) > \mathrm{dist}(x_i, x_j)$	$A_{21}(i,j)$	$A_{22}(i,j)$

This is the 2×2 contingency table that we use to conduct a classic Pearson's test of independence. Specifically, this statistic can be written as:

$$S(i,j) = \frac{(N-2)\{A_{12}(i,j)A_{21}(i,j) - A_{11}(i,j)A_{22}(i,j)\}^2}{A_{1\cdot}(i,j)A_{2\cdot}(i,j)A_{\cdot 1}(i,j)A_{\cdot 2}(i,j)}$$

where $A_{1\cdot}(i,j) = A_{11}(i,j) + A_{12}(i,j)$ and so on. $S(\cdot,\cdot)$ is calculated for every pair of points in the sample. Then, as stated before, the overall test statistic is the sum of all the statistics $S(\cdot,\cdot)$. The distribution of this statistic is unknown, and therefore a permutation method is required to calculate its p-value. This will be explained in the methodology section.

In summary, for X and Y that are vectors, we cannot visualize easily the zones centered at (x_i, y_i) and delimited by the radii R_{x_0}, R_{y_0}. However, the idea of the test stays the same: if for some of those zones knowledge about being inside (or outside) the zone according to the X coordinate gives some knowledge about being inside (or outside) the zone according to the Y coordinate, then independence does not hold.

Chapter 3

HHG ON RANKED DATA

3.1. Motivation

In this section we propose to use the HHG test on ranks of the observed data, rather than on the data itself. That is, if we have a sample of size N from the joint distribution of (X, Y), say

$$(X_1, Y_1), \cdots, (X_N, Y_N),$$

we propose to compute the ranks of X and Y, call them R and S respectively, yielding

$$(R_1, S_1), \cdots, (R_N, S_N),$$

and then to apply the HHG test on this set of ranks (R, S). To give an example of what the rank transformation is, say we have the following sample of size five from a univariate random variable X:

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{pmatrix} = \begin{pmatrix} 4 \\ 5 \\ 6 \\ 2 \\ 10 \end{pmatrix}.$$

Under the rank transformation we obtain:

$$\begin{pmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 4 \\ 1 \\ 5 \end{pmatrix}.$$

This is a very simple modification to the original procedure in Heller et al. [15] that we believe improves its power in many situations. Moreover, we believe this change makes the test more robust. Here we give some motivation (and visual examples) for this change in the case where X and Y are random variables, however this modified version works for random vectors as well, see an example at the end of this section.

Now, why do we do such a transformation? For starters, note that for continuous random variables X and Y we have

$$X \perp Y \iff F_X(X) \perp F_Y(Y).$$

A proof of this statement is given in Appendix A.2. This means that, at least for continuous random variables, investigating the independence between X and Y is the same problem as investigating the independence between $F_X(X)$ and $F_Y(Y)$. Furthermore, and perhaps most importantly, this transformation puts the observations on the same scale. This is suitable when trying to detect dependence. Indeed, when plotting X against Y, it is not always straightforward to identify relationships between variables because elements such as skewness, spread or extreme values might make it look like there is a relationship when in fact there is none. Likewise, when variables are not on the same scale, it might be harder to identify a relation that is present. Of course, with data that we have collected, we do not know the 'true' functions $F_X(x)$ and $F_Y(y)$, so instead we use their empirical counterparts, noted $F_N(x)$ and $F_N(y)$. But those are only the ranks divided by N, that is:

$$F_N(x_i) = \frac{1}{N}\sum_{j=1}^{N} \mathbb{1}_{x_j \leq x_i} = \frac{R_i}{N}.$$

So to summarize, if $F_X(X)$ and $F_Y(Y)$ are dependent it means X and Y are dependent. Because $F_X(X)$ and $F_Y(Y)$ are unknown we use their empirical counterparts $F_N(x)$ and $F_N(y)$, which are based on the ranks of X and Y. Hence, what we do in the end is trying to detect dependence between the ranks of our observations.

Note that for eventual tied values of the random variables X or Y we use the 'average' method to assign ranks. This means we assign to those tied observations the same rank, with value equal to the average of the ranks we would have without ties. For instance, the sequence {2, 4, 5, 5, 10} would become $\{1, 2, 3.5, 3.5, 5\}$ under the rank transformation. Likewise the sequence {2, 2, 5, 5, 10} would become {1.5, 1.5, 3.5, 3.5, 5}.

We now give an example of what we mean when we say independence (or dependence) is easiest to see on ranked data. Consider Figure 3.1. It shows the scatter plot of two perfectly independent random variables, call them X and Y, both having an Exponential distribution of mean 2. It is difficult to conclude if there is a relation between the two because of the highly asymmetrical way the observations are distributed on their support (here roughly 0 to 12). On the contrary, if we plot the empirical distribution function of the observations $F_N(X)$ and $F_N(Y)$ against one another we obtain a much clearer picture. Indeed, Figure 3.2 shows no pattern whatsoever.

As another example, consider two independent random variables $Z \sim \mathrm{N}(2, \sqrt{12})$ and $W \sim \text{Pareto}(\alpha = 3, \theta = 4)$. They both have the same mean (2) and variance (12), however the Normal is light-tailed while the Pareto is heavy-tailed. Hence, if we look at the scatter plot of W against Z on Figure 3.3, it is not straightforward to exclude a relation between the two. Again, plotting the empirical cdf yields a clear picture: on Figure 3.4 we cannot detect any relation between $F_N(Z)$ and $F_N(W)$.

In one last example, let us consider for a change two *dependent* random variables. The first one is called C and has a Cauchy distribution with scale parameter 3. The second one is called S and is defined as $S = \sin(C) + \mathrm{N}(0, \frac{1}{2})$. If we take a look at the scatter plot of S against C (figure 3.5), it is not very straightforward to establish there is a relation. This is because the Cauchy distribution generates very 'extreme' values, which distorts the plot: the X axis goes to 200 only because of a couple of points, which squeezes all the other points, making it hard to see a pattern. On the contrary, looking at the scatter plot of the ecdf $F_N(S)$ vs $F_N(C)$ as displayed in Figure 3.6, we easily detect a pattern, hence rejecting independence.

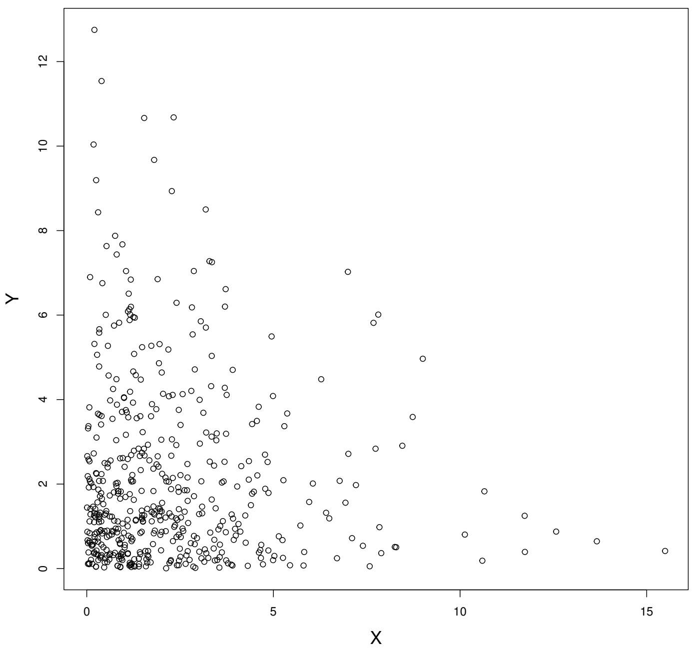

FIG. 3.1. Scatter plot of two independent Exp(1/2)

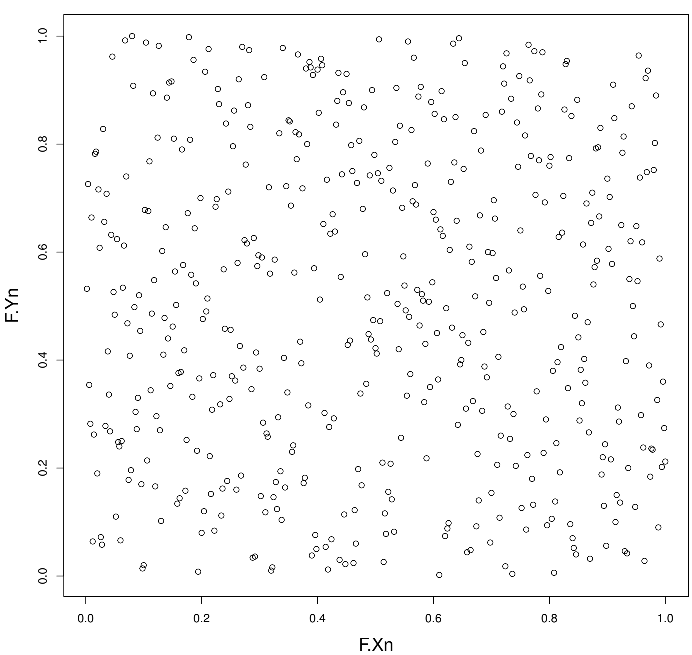

FIG. 3.2. Scatter plot of the ecdf of two independent Exp(1/2)

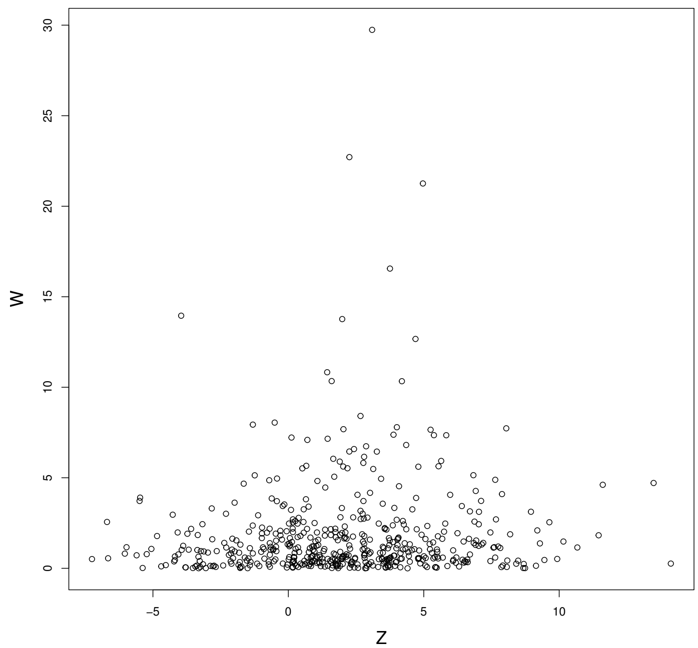

FIG. 3.3. Scatter plot of a Pareto($\alpha = 3, \theta = 4$) against an independent N($2, \sqrt{12}$)

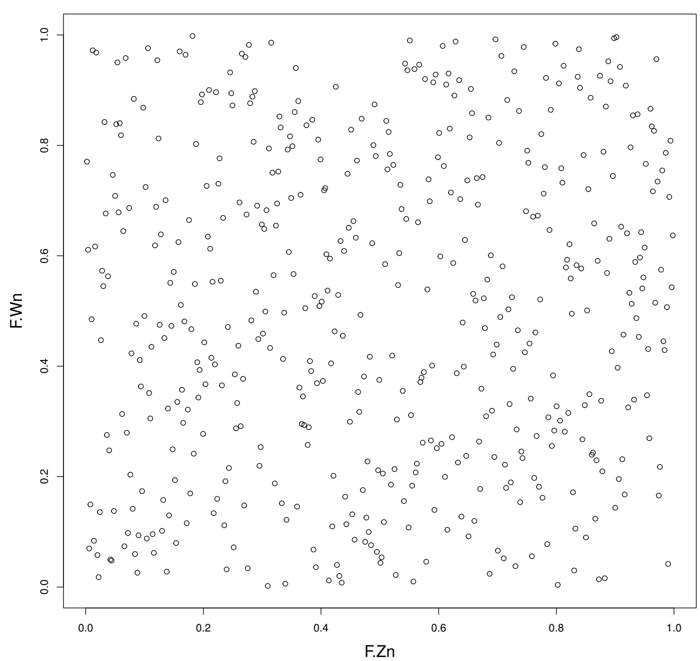

FIG. 3.4. Scatter plot of the ecdf of a Pareto($\alpha = 3, \theta = 4$) against the ecdf of an independent N($2, \sqrt{12}$)

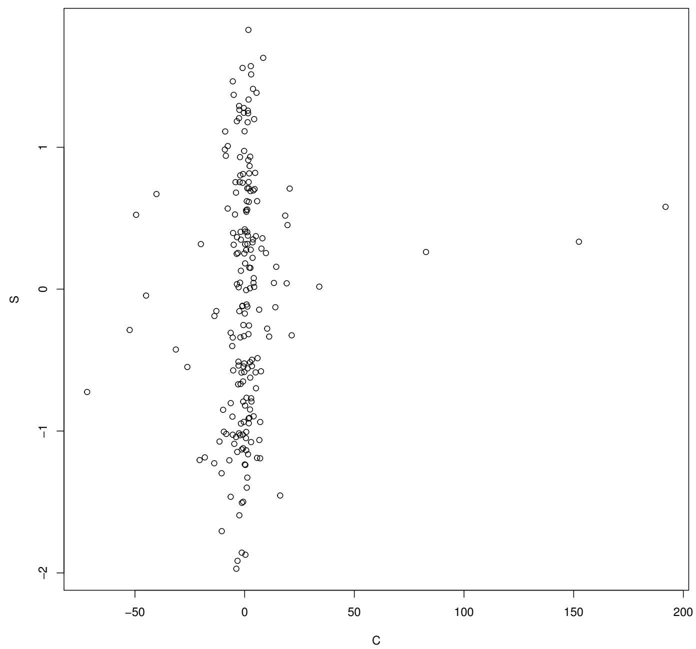

FIG. 3.5. Scatter plot of a C against $\sin(C)$ + N(0, 0.5)

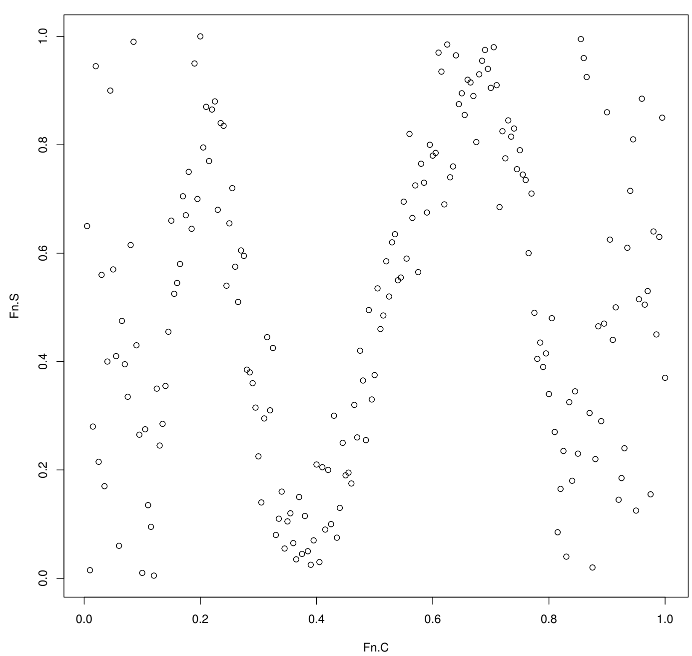

FIG. 3.6. Scatter plot of the ecdf of C against the ecdf of $\sin(C)$ + N(0, 0.5)

Finally, note that if X and Y are vectors, we can still apply the rank transformation to them and then perform the HHG test on the transformed data. Precisely, for each of the components of X and Y we replace the original values by their within-component ranks (values from 1 to N).

To give an example, say X has two components, $X = (X^{(1)}, X^{(2)})$, and we have a sample of size five from X, for instance:

$$\begin{pmatrix} X_1^{(1)} & X_1^{(2)} \\ X_2^{(1)} & X_2^{(2)} \\ X_3^{(1)} & X_3^{(2)} \\ X_4^{(1)} & X_4^{(2)} \\ X_5^{(1)} & X_5^{(2)} \end{pmatrix} = \begin{pmatrix} 4 & 14 \\ 5 & 6 \\ 6 & 8 \\ 2 & 22 \\ 10 & 3 \end{pmatrix},$$

then under the rank transformation we obtain:

$$\begin{pmatrix} R_1^{(1)} & R_1^{(2)} \\ R_2^{(1)} & R_2^{(2)} \\ R_3^{(1)} & R_3^{(2)} \\ R_4^{(1)} & R_4^{(2)} \\ R_5^{(1)} & R_5^{(2)} \end{pmatrix} = \begin{pmatrix} 2 & 4 \\ 3 & 2 \\ 4 & 3 \\ 1 & 5 \\ 5 & 1 \end{pmatrix}.$$

3.2. p-value of the test

In the original HHG test, as well as in our version using ranks, we do not know the distribution of the test statistic under H_0. Asymptotically, the $S(\cdot, \cdot)$ statistics are χ^2, but they are dependent, hence their sum does not have (or at least we have not found) a known distribution. We then rely on a permutation method to compute the p-value of this test statistic.

Suppose we have calculated the test statistic of a sample to be t_{obs}. On a qualitative level, the bigger t_{obs} is, the stronger is the evidence to reject H_0. Recall the basic definition of the p-value:

$$p_t = P[T \geq t | H_0].$$

In words, this is the probability under H_0 to obtain a test statistic T equal to or "more extreme" to the value t we indeed observed. Because we do not know the distribution of T, we can't compute this probability exactly. However, it is possible to generate a sample of T under H_0, noted $t_1^*, t_2^*, ...$ and use it to calculate an empirical p-value.

Note that if H_0 is true and there is no association of any form between X and Y then the occurrence of a variable has no impact on the occurrence of the others, and vice-versa. Hence, if we reshuffle all the observations within the sample, it would make no material impact on T, our measure of dependence.

In other words, a reshuffled sample imitates a sample generated under H_0, without changing the marginals of X and Y. Therefore a test statistic calculated on this reshuffled sample has the distribution of T under H_0. This means we can have a good approximation of p_t if we apply the following procedure.

(1) Choose the number of permutations n_p (a "big" number such as 1000).
(2) Calculate t_{obs}, the test statistic based on the original sample.
(3) Generate a sample from the original data by randomly permutating the rows of Y, where a row represents one sample point.
(4) Calculate the test statistic based on this new sample. Call it t_1^*.
(5) Repeat steps (3) and (4) n_p times.
(6) Count the number of times $t_i^* \geq t_{obs}$. Call it m.
(7) Calculate the p-value as $p_t = m/n_p$.

3.3. Power simulations

3.3.1. How to estimate the power

We now want to estimate the power of our test:

$$\textit{power} = \mathrm{P}[\text{reject } H_0 | H_0 \text{ is false }].$$

Of course, such a probability depends on the way H_0 is false, i.e. it depends on the form of the alternative H_1. Hence we proceed with simulations. We follow this procedure:

(1) Choose a dependence structure (i.e. fix H_1).
(2) Generate a set of data under H_1.
(3) Apply the test to this data, and record if H_0 is rejected.
(4) Repeat this B times, where B is a "big" number such as 10,000.

The power is then given by:

$$\textit{power} = \frac{\#\text{ times } H_0 \text{ is rejected}}{B}.$$

However, when doing simulations, because we fix H_1, we know precisely what are the marginals of the random vectors X and Y. Hence, the test no longer requires us to use a permutation method to obtain an independent sample under H_0 and use it to calculate the p-value of our statistic T. Rather, by simulation we can generate such samples, and then obtain the required quantiles of T under the null hypothesis H_0. Then, for each of the B iterations in the above algorithm, we reject H_0 if the calculated test statistic T is bigger then this (empirical) quantile. In other words, for a specific H_1 we follow the steps:

(1) Generate a sample with independent X and Y having marginals specified by H_1.
(2) Calculate the test statistic T^* based on this sample.
(3) Repeat this M times, where M is a "big" number such as 50,000.
(4) Use the resulting sample $t_1^*, \cdots, t_M^*$ to calculate the empirical. $(1-\alpha)\%$-quantile of T_0:

$$t_{1-\alpha}^*.$$

(5) Then generate B samples under H_1 and each time, use the following rule to reject H_0:

$$\text{reject } H_0 \text{ if } T \geq t_{1-\alpha}^*.$$

Note that because this procedure is computationally intensive, coding in pure R would be too slow and we use instead the R package 'Rcpp' by Eddelbuettel et al. [8] which allows the integration of C coding into R, the execution of C code being significantly faster than R to perform loops, see Lafaye De Micheaux et al. [20].

3.3.2. Comparison to other tests

Computing the power in many different dependence situations, both univariate (random variables) and multivariate (random vectors) will give us information about how well the test performs. However, we also want to compare our test to other existing procedures. Hence, in the next sections 3.3.3 and 3.3.4, we compare the power of our ranked-version of the test in Heller et al. [15] (labeled BLAW) to three other independence tests: the original test from Heller et al. [15] (HHG), the distance covariance test (DCOV) from Székely et al. [34] and the Hilbert-Schmidt independence criterion (HSIC) from Gretton et al. [14]. Many different dependence structures are investigated. First, in sub section 3.3.3 we explore examples where X and Y are random variables (both have one component). Next, in sub section 3.3.4 we explore examples where X and Y are vectors, both of dimension two: $X = (X_1, X_2)$, $Y = (Y_1, Y_2)$.

For each example, we first state what the dependence structure is, then we present graphs of the empirical power (in %) against the sample size (usually from 10 to 100 by leaps of 10). We use a level $\alpha = 5\%$ in all examples. All of the results are based on 50,000 simulations to estimate the $(1-\alpha)\%$-quantile of $T|H_0$, and 10,000 power simulations. Hence $M = 50,000$ and $B = 10,000$ in the notation of section 3.3.1. The discussion of the results is deferred to section 3.4.

3.3.3. Random variables examples

First we consider the first six examples in table 3 from Heller et al. [15], labeled 'four independent clouds' (this is H_0), 'W-shape', 'Diamond', 'Parabola', 'Two parabolas', and 'Circle'. Note that we changed the original examples slightly because a deterministic sequence was used in the R code that generated them:

```
seq(-1, 1, length = n)
```

Making the variables not i.i.d. We used instead:

```
runif(n, -1,1)
```

In addition to those examples, we consider three dependence models where Y is a function of X (with an additional noise so that the dependence is not picked-up too easily by the tests):

- Linear dependence: $X \sim N(0,1),\ \ Y = X + N(0, \frac{1}{2})$
- Exponential dependence: $X \sim U(-3,3),\ \ Y = \exp(X/3) + U(-3,3)$
- Sine dependence: $X \sim U(0, 2\pi),\ \ Y = \sin(X) + N(0,1)$

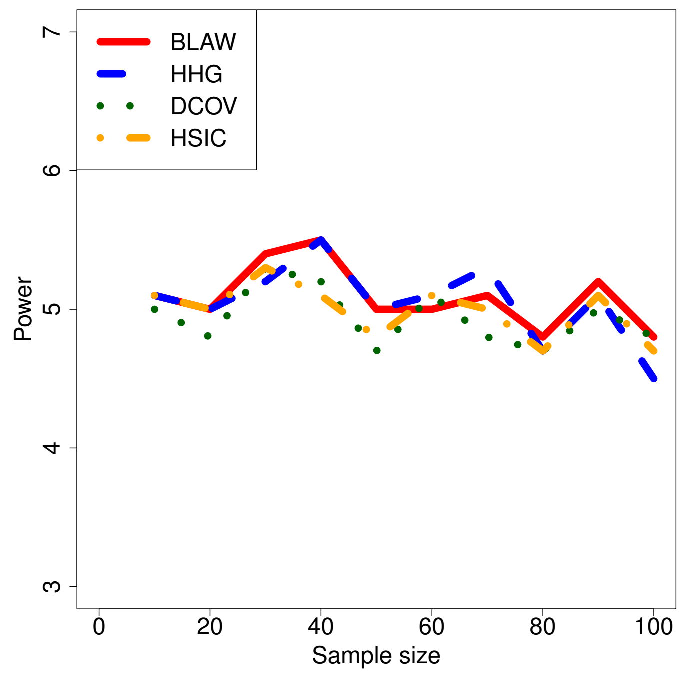

FIG. 3.7. Power in the case of four independent clouds

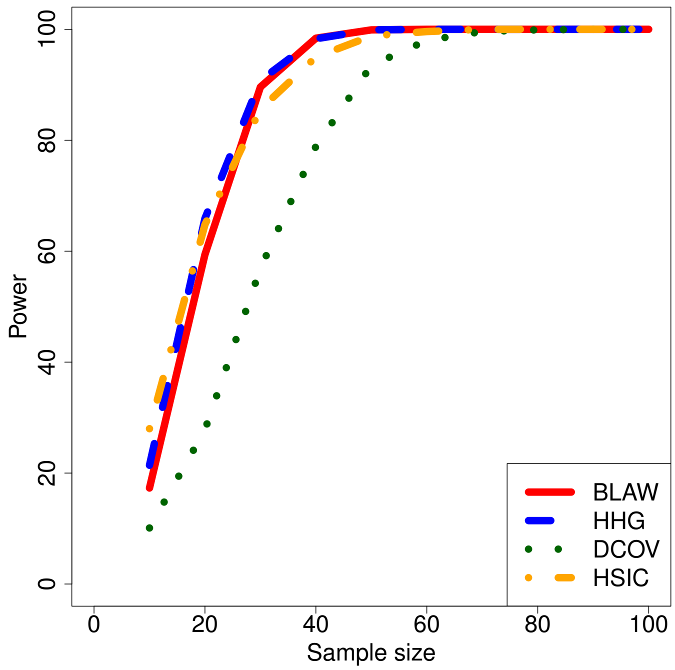

FIG. 3.8. Power in the case of the W-shape dependence

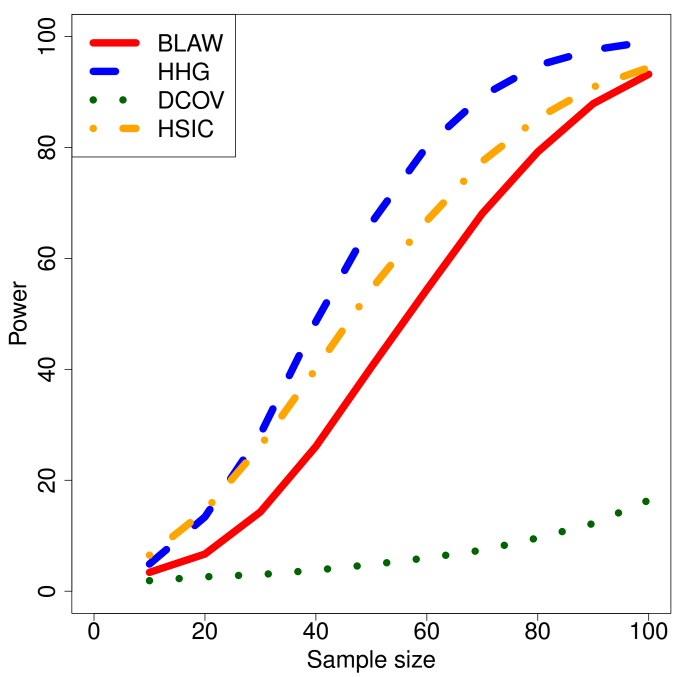

FIG. 3.9. Power in the case of the Diamond dependence

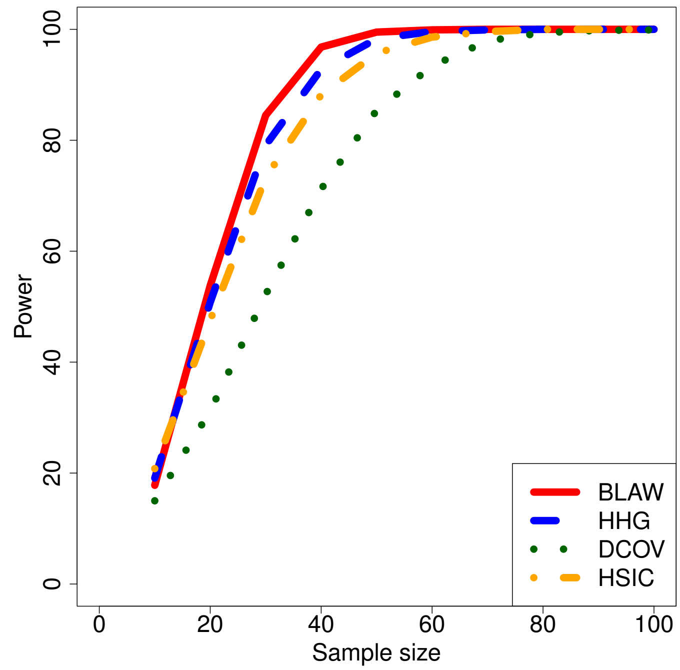

FIG. 3.10. Power in the case of the Parabola dependence

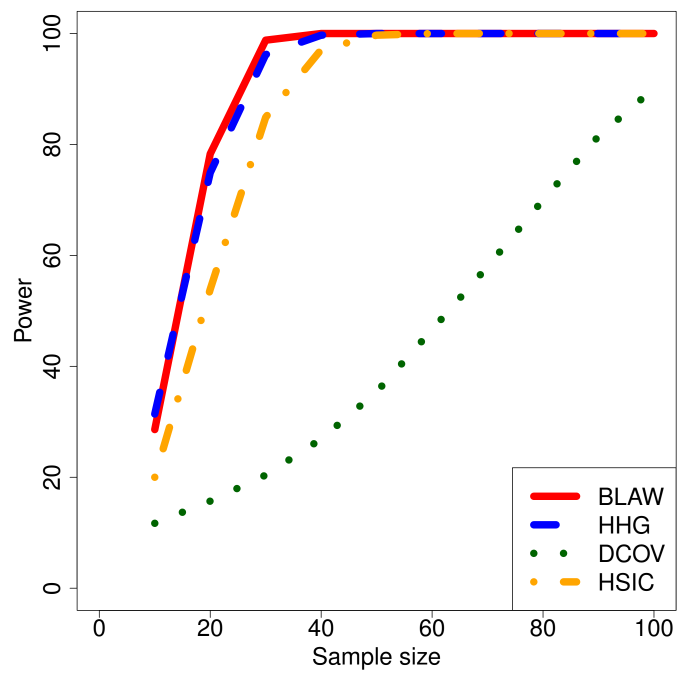

FIG. 3.11. Power in the case of the Two Parabolas dependence

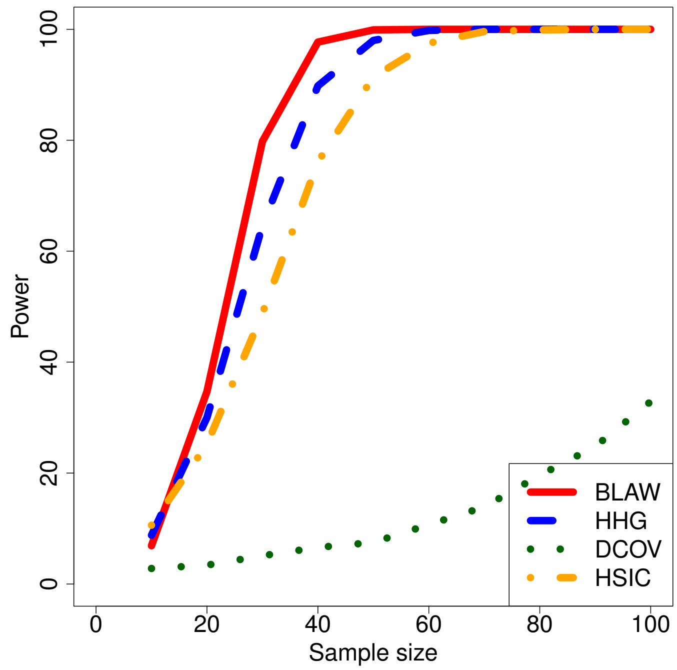

FIG. 3.12. Power in the case of the Circle dependence

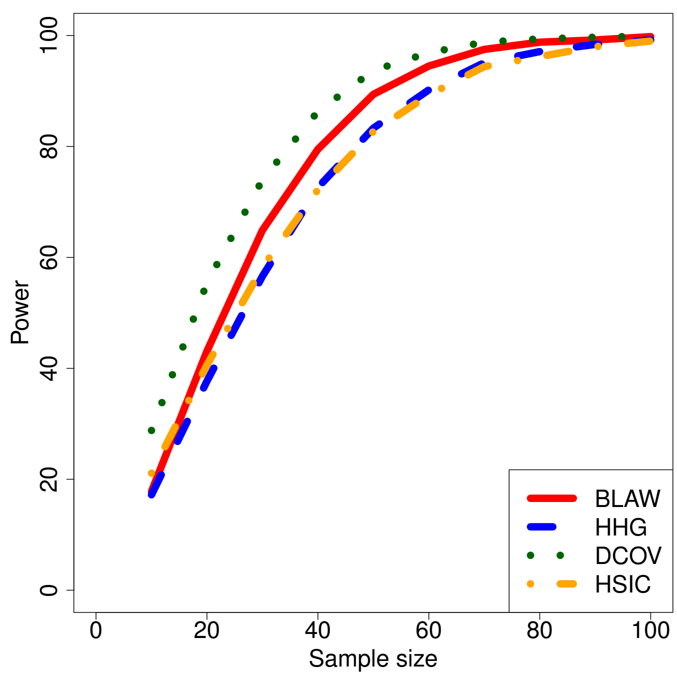

FIG. 3.13. Power in the case of the Linear dependence

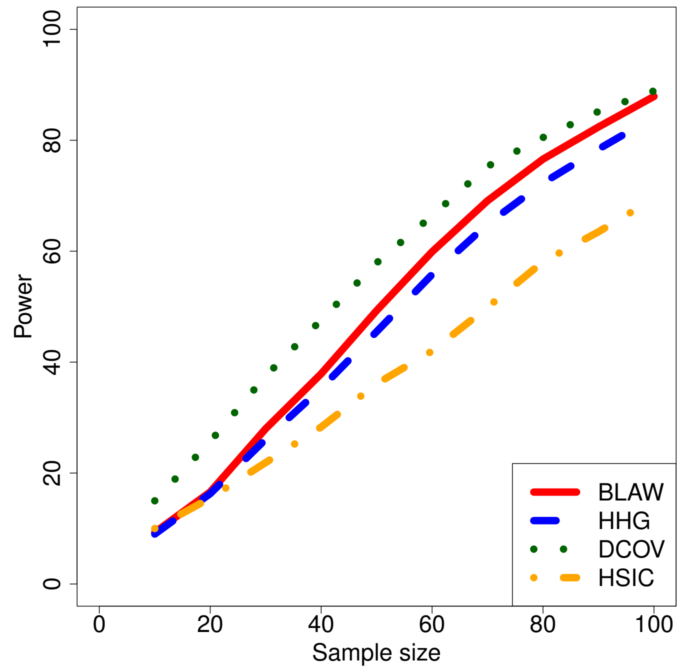

FIG. 3.14. Power in the case of the Exponential dependence

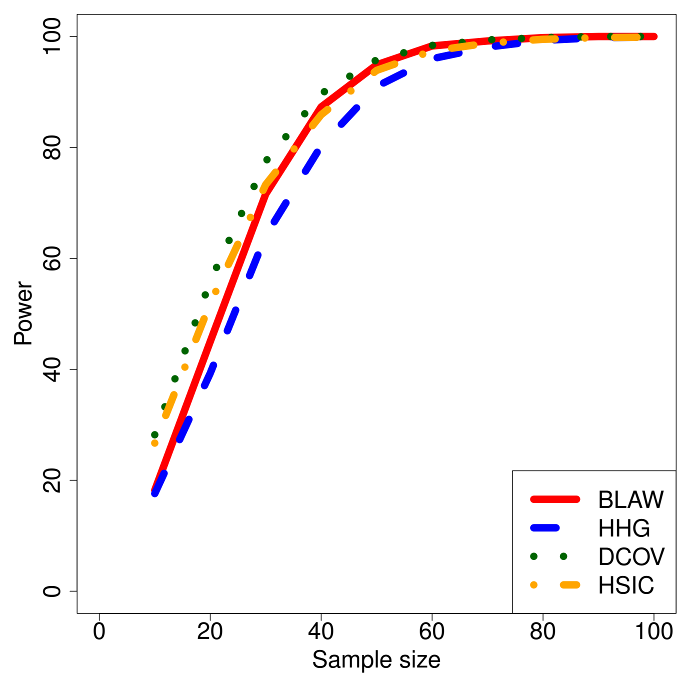

FIG. 3.15. Power in the case of the Sine dependence

In the next examples, we use copulas to generate dependent data. Copulas have the advantage that we can change the margins of the random variables, without changing the strength of the dependence. Hence, we can investigate if the power of the different methods varies according to the margins. As in Genest and Rémillard [10] we consider the three following copulas:

- Clayton
- Gumbel
- Normal

And we use three different margins, for a total of nine examples:

- Normal $(0, 1)$
- Exponential (scale $= 1$)
- Cauchy (scale $= 1$)

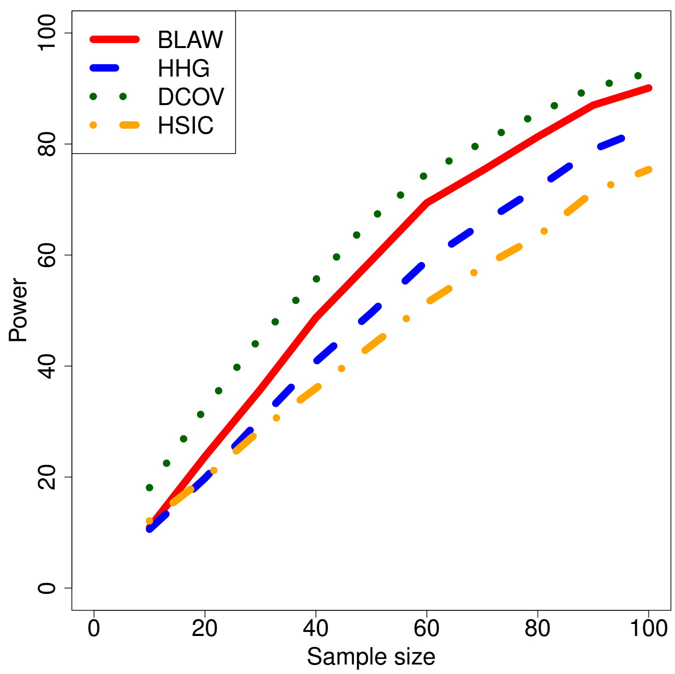

FIG. 3.16. Power in the case of the Clayton ($\theta = 0.6$) copula with Normal margins

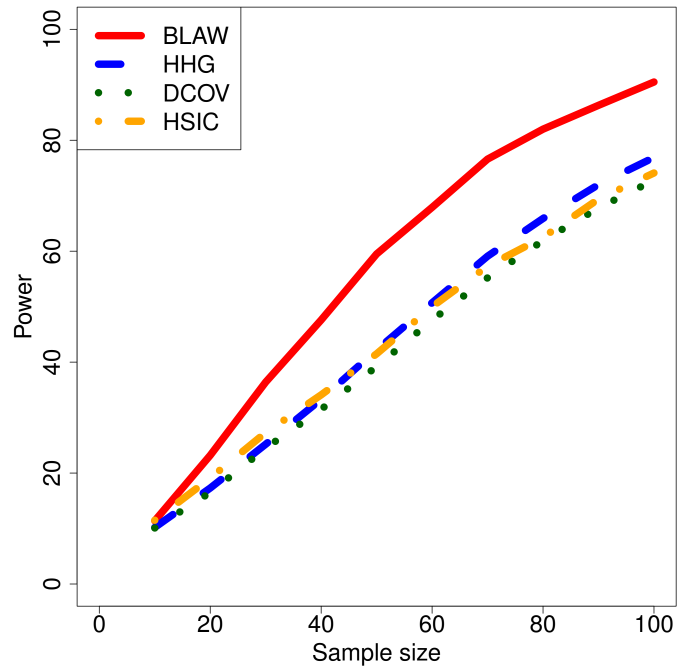

FIG. 3.17. Power in the case of the Clayton ($\theta = 0.6$) copula with Exponential margins

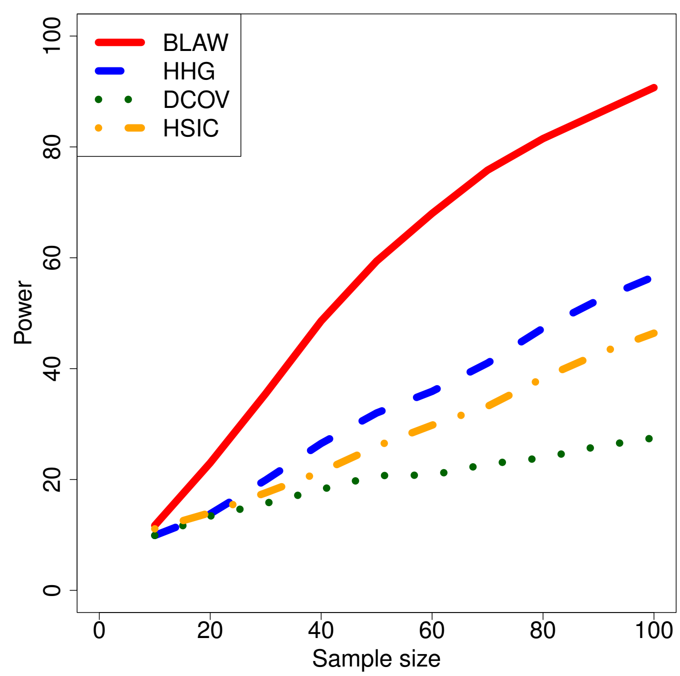

FIG. 3.18. Power in the case of the Clayton ($\theta = 0.6$) copula with Cauchy margins

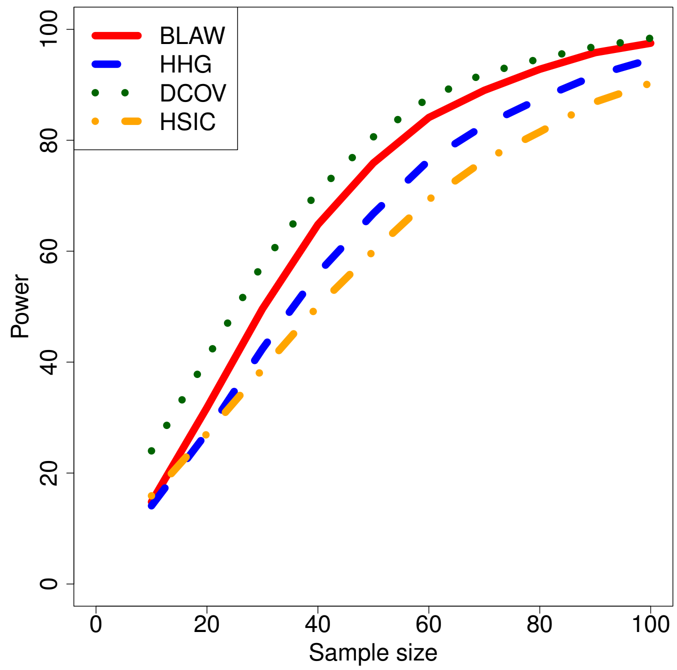

FIG. 3.19. Power in the case of the Gumbel ($\theta = 1.4$) copula with Normal margins

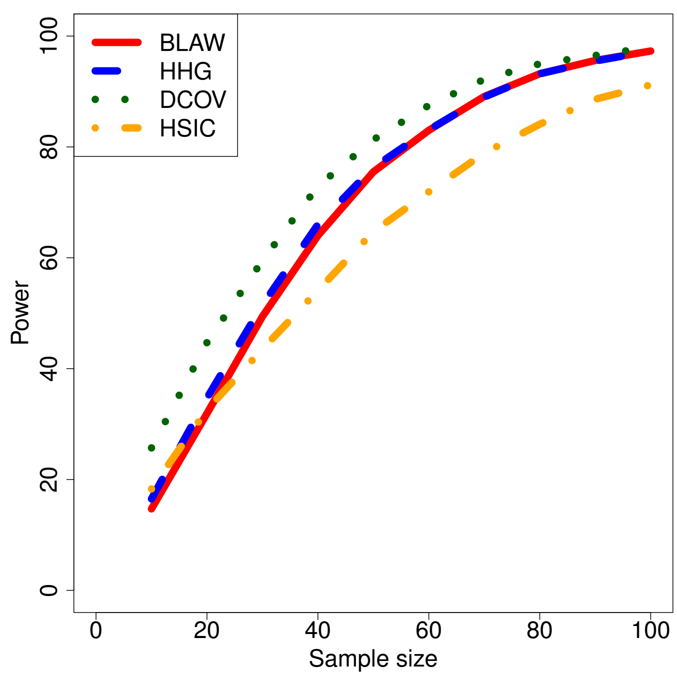

FIG. 3.20. Power in the case of the Gumbel ($\theta = 1.4$) copula with Exponential margins

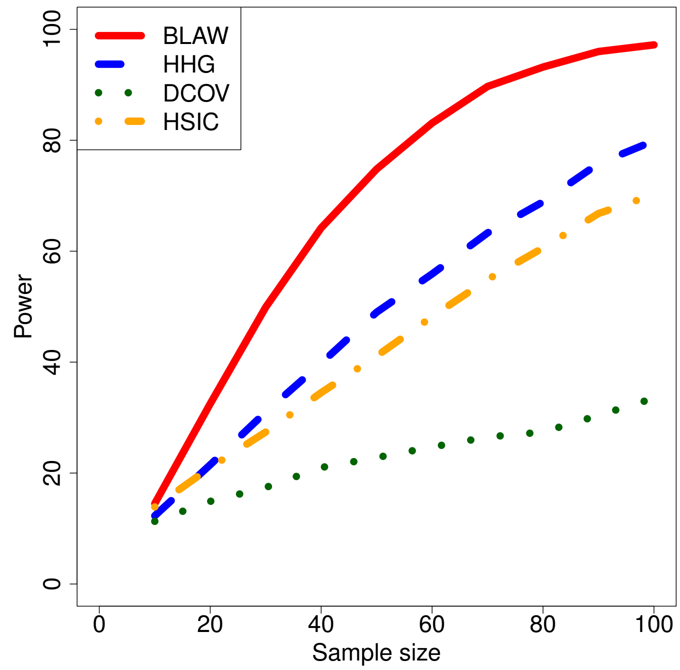

FIG. 3.21. Power in the case of the Gumbel ($\theta = 1.4$) copula with Cauchy margins

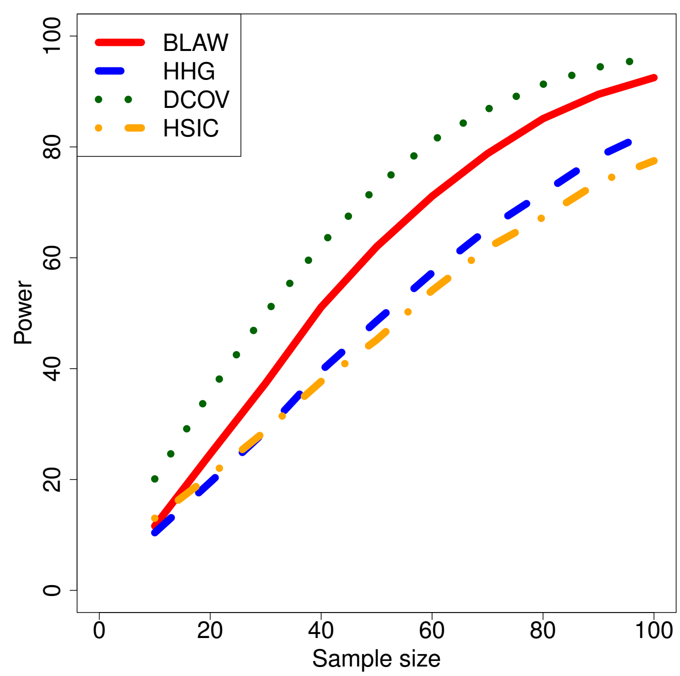

FIG. 3.22. Power in the case of the Normal ($\rho = 0.4$) copula with Normal margins

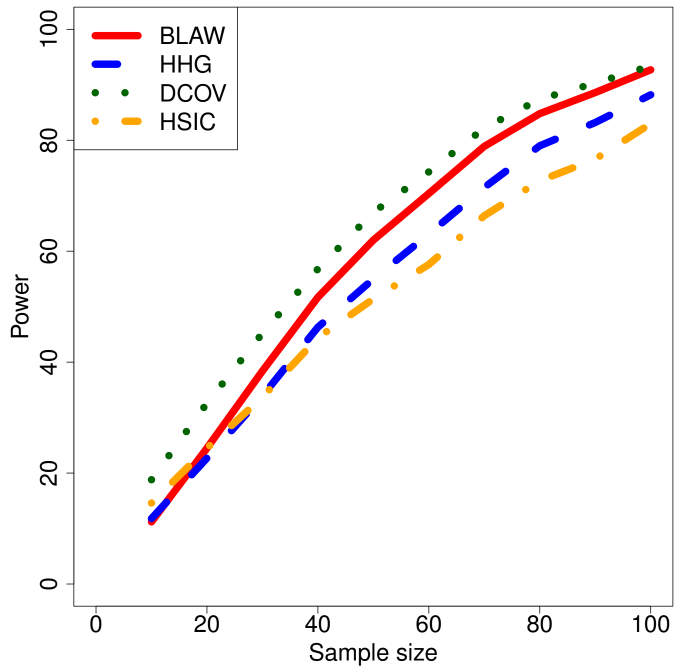

FIG. 3.23. Power in the case of the Normal ($\rho = 0.4$) copula with Exponential margins

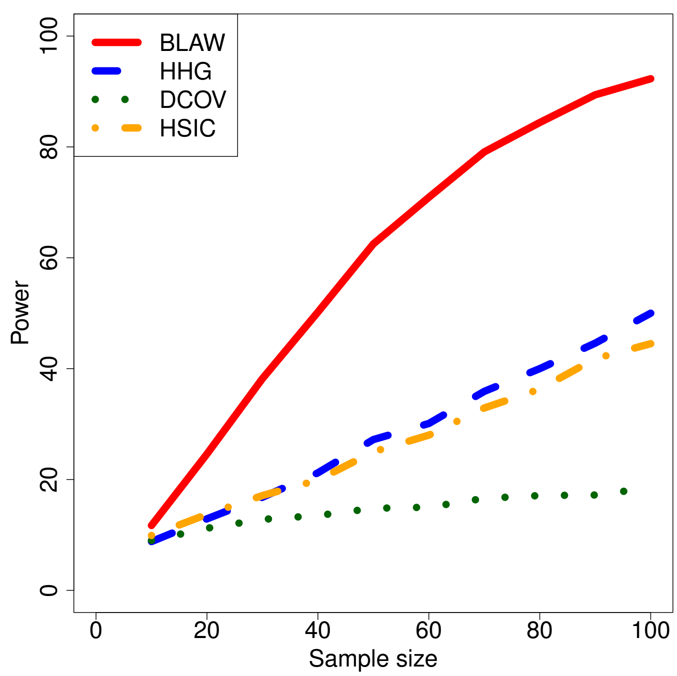

FIG. 3.24. Power in the case of the Normal ($\rho = 0.4$) copula with Cauchy margins

3.3.4. Random vectors examples

Here we consider first three examples where X and Y are 5-dimensional vectors. We consider the last two examples in table 3 from Heller et al. [15], as well as the example from table 4 with $\beta_1 = 1, \beta_2 = 4$. We label those examples the "logarithmic", "epsilon" and "quadratic" dependences. Specifically, the models are as specified below.

- Logarithmic dependence:

$$X_i \sim N(0,1), \qquad \forall i \in \{1,2,\cdots 5\}$$
$$Y_i = \log(X_i^2), \qquad \forall i \in \{1,2,\cdots 5\}$$

- Epsilon dependence:

$$X_i \sim N(0,1), \qquad \forall i \in \{1,2,\cdots 5\}$$
$$\epsilon_i \sim N(0,1), \qquad \forall i \in \{1,2,\cdots 5\}$$
$$Y_i = X_i \cdot \epsilon_i, \qquad \forall i \in \{1,2,\cdots 5\}$$

- Quadratic dependence:

$$X_i \sim N(0,1), \qquad \forall i \in \{1,2,\cdots 5\}$$
$$\epsilon_i \sim N(0,3), \qquad \forall i \in \{1,2,\cdots 5\}$$
$$Y_i = \begin{cases} X_i + 4X_i^2 + \epsilon_i, & \text{for } i \in \{1,2\} \\ \epsilon_i, & \text{for } i \in \{3,4,5\} \end{cases}$$

We add two more examples to those. One is based on our motivational example in section 1.3.2, and we call it the "2D pairwise independence" example.

- 2D pairwise independence:

$$X, \text{ and } Z_0 \sim N(0,1)$$
$$Y_1 \sim N(0,1), \quad Y_2 = |Z_0| \cdot \text{sign}(X \cdot Y_1)$$

Hence X is a random variable and Y is a random vector with two components. Note this is a rather odd construction: X, Y_1 and Y_2 are all pairwise independent, but not mutually independent, hence a universally consistent test between X and (Y_1, Y_2) should reject H_0 in this situation.

The other example, which we call "big noise" is the following.

- Big noise:

$$X_1 \sim N(0,1) \text{ and } X_2 \sim N(0,4)$$

$$Y = X_1^2 + N(0, 2)$$

We call this example 'big noise' because here X_2 is independent of Y and has a big standard deviation compared to X_1 (which is dependent of Y). Hence, when trying to detect dependence between the vector (X_1, X_2) as a whole and the variable Y, the noise component X_2 makes the task harder.

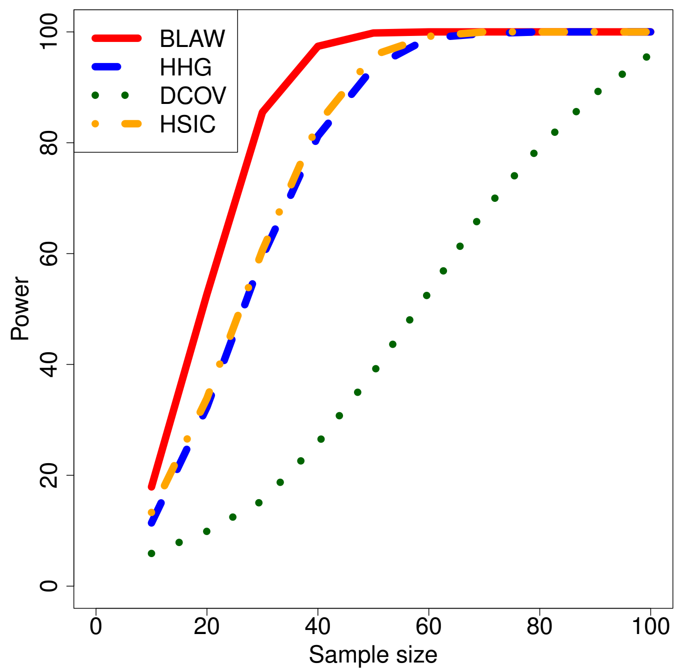

FIG. 3.25. Power in the case of the 'log' dependence

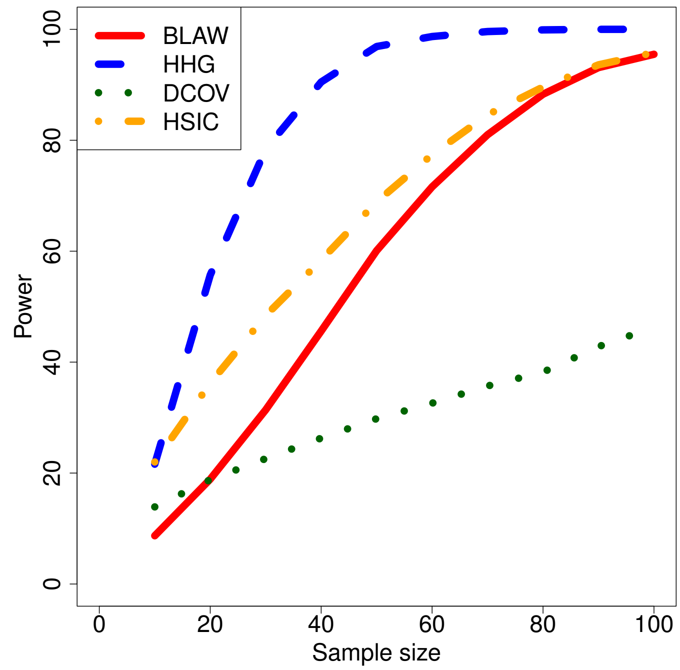

Fig. 3.26. Power in the case of the 'epsilon' dependence

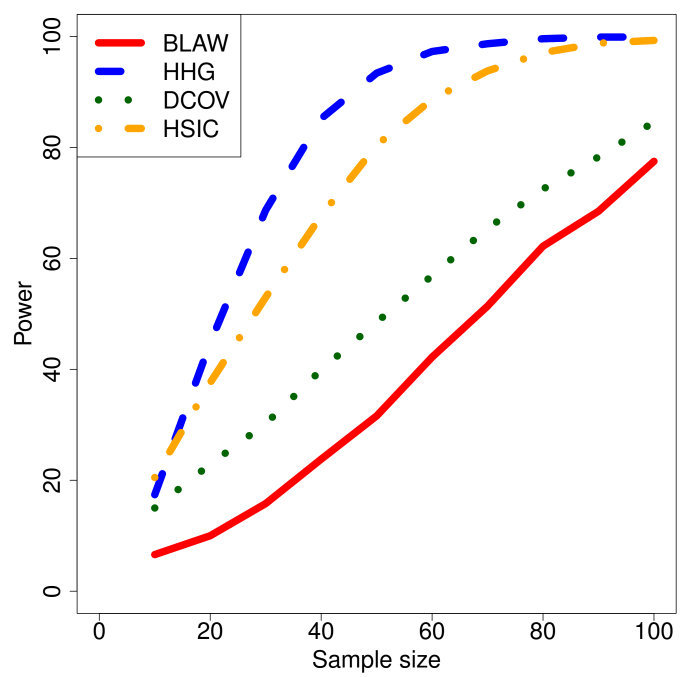

Fig. 3.27. Power in the case of the 'quadratic' dependence

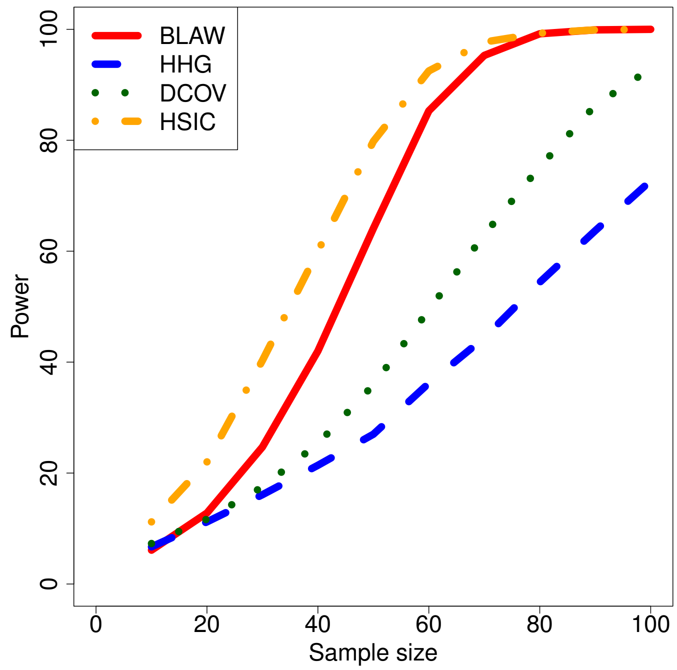

FIG. 3.28. Power in the case of the 'pairwise independence' example

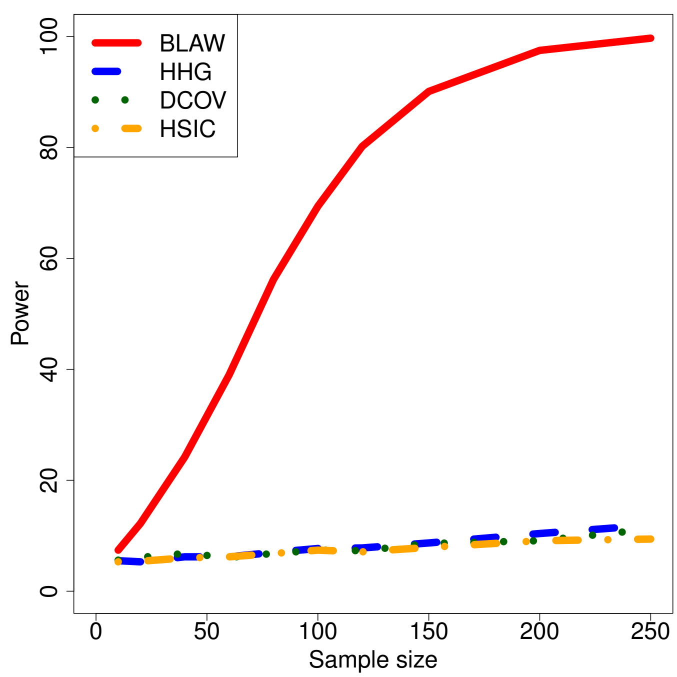

FIG. 3.29. Power in the case of the 'big noise' example

Next, we consider dependent data via copula structures. Here X and Y are both of dimension two. Again we consider three copulas (Clayton, Gumbel, Normal) with three marginals (Normal, Exponential, Cauchy), yielding nine examples. Note that in the case of the Normal copula, we set the correlations between all pairs of variables to 0.3.

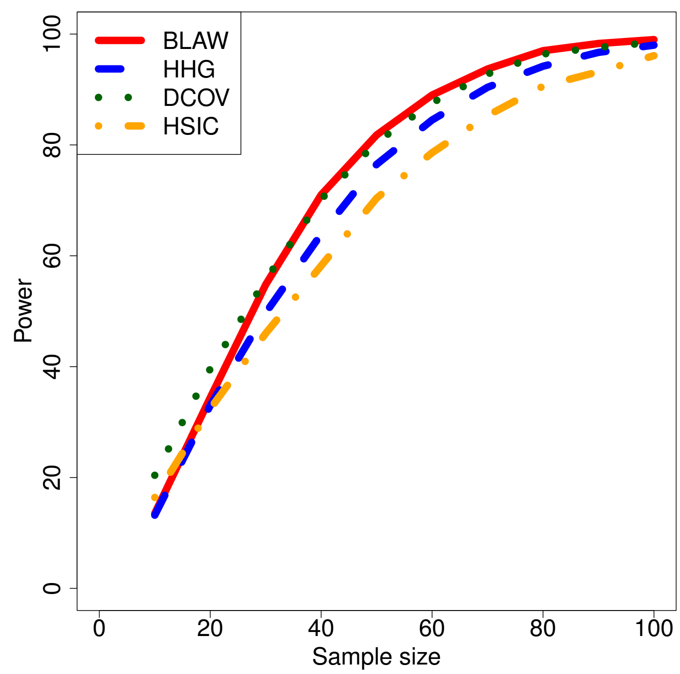

FIG. 3.30. Power in the case of vectors for the Clayton (0.5) copula with Normal margins

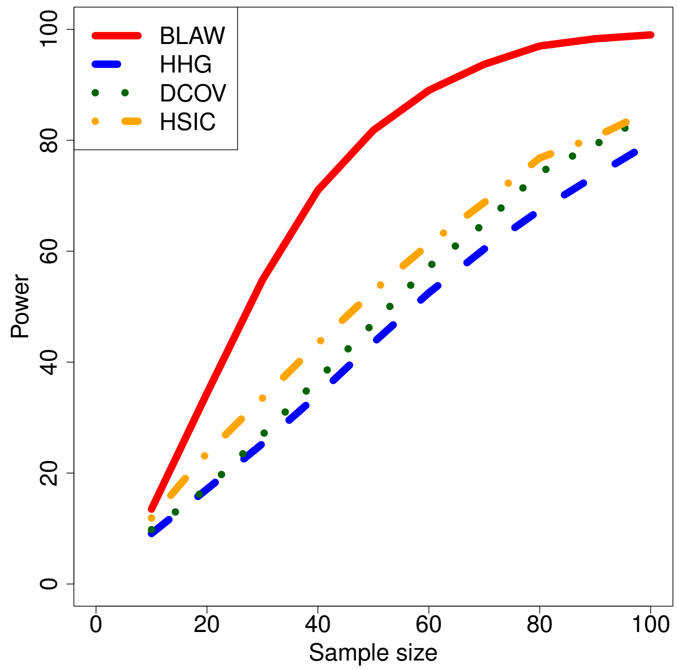

FIG. 3.31. Power in the case of vectors for the Clayton (0.5) copula with Exponential margins

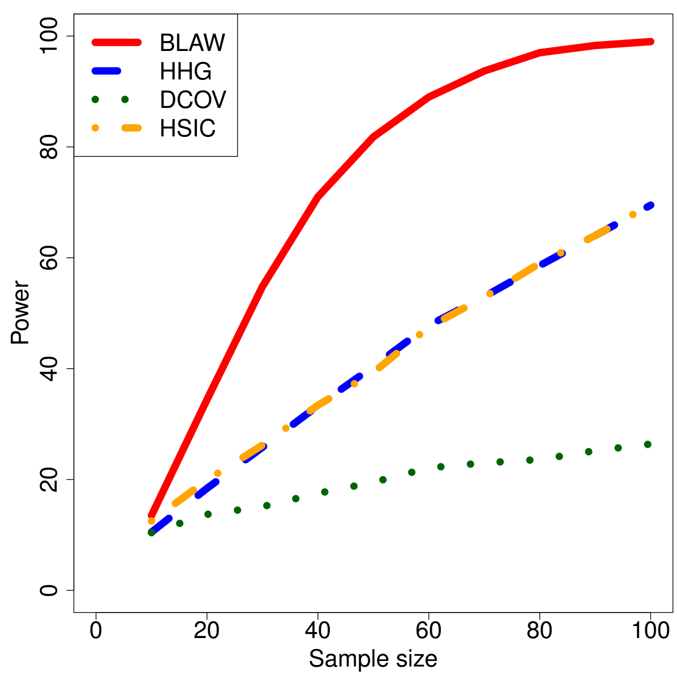

FIG. 3.32. Power in the case of vectors for the Clayton (0.5) copula with Cauchy margins

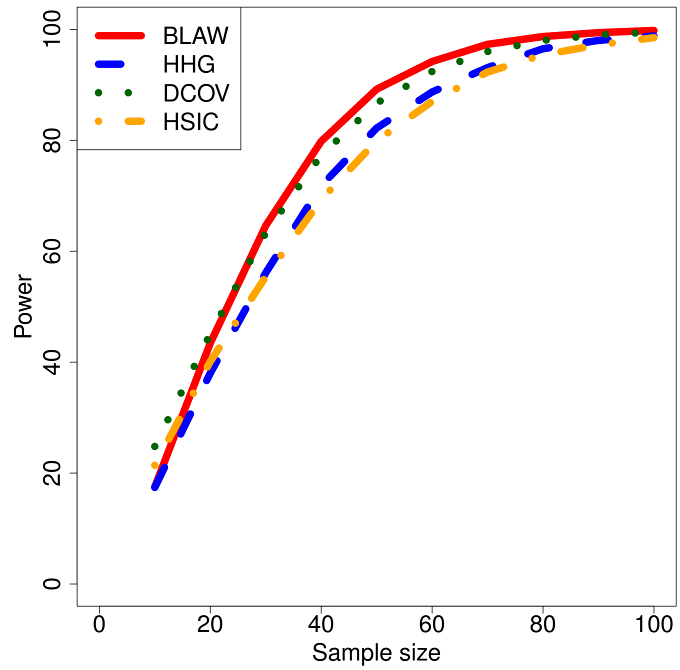

FIG. 3.33. Power in the case of vectors for the Gumbel (1.3) copula with Normal margins

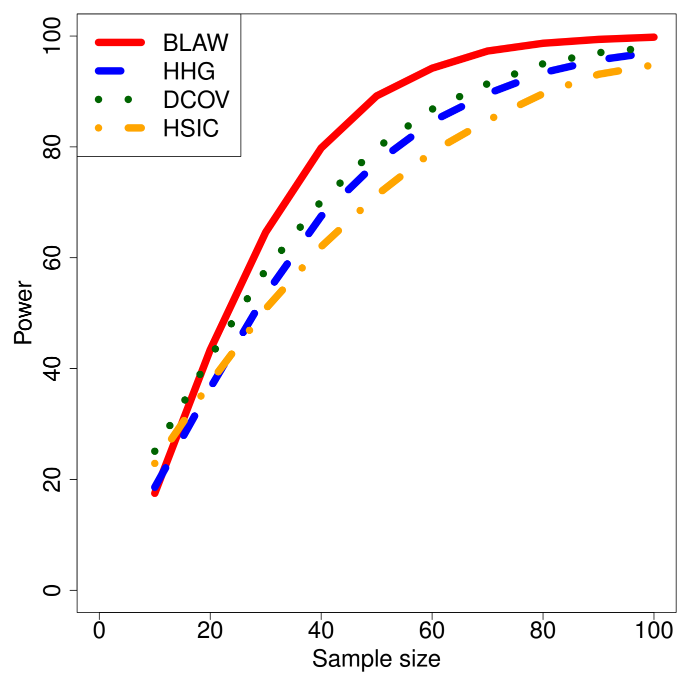

FIG. 3.34. Power in the case of vectors for the Gumbel (1.3) copula with Exponential margins

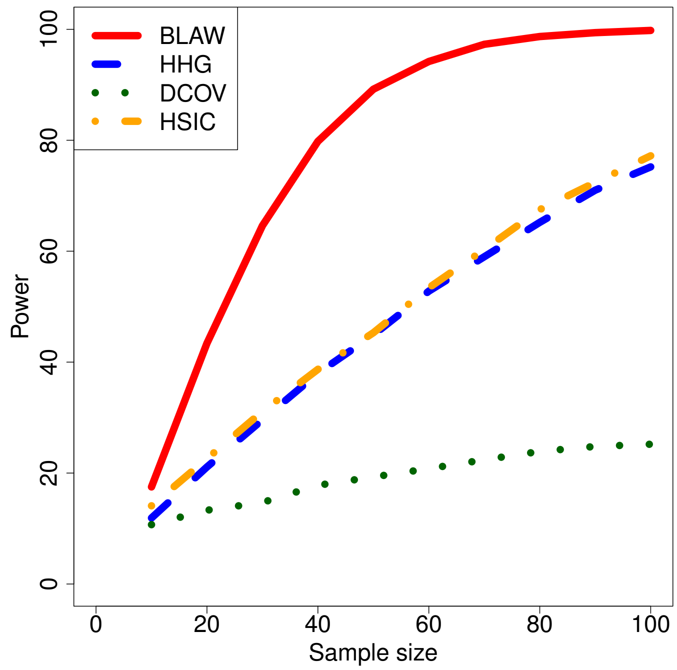

FIG. 3.35. Power in the case of vectors for the Gumbel (1.3) copula with Cauchy margins

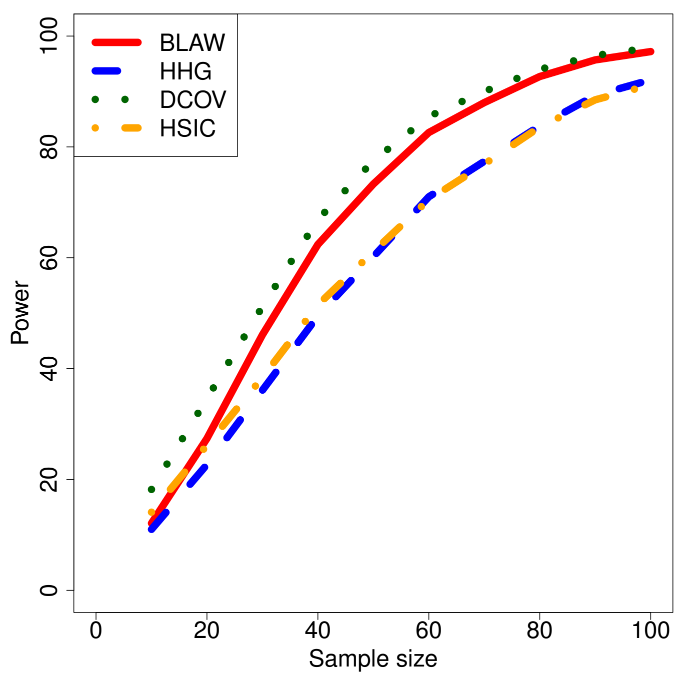

FIG. 3.36. Power in the case of vectors for the Normal ($\rho = 0.3$) copula with Normal margins

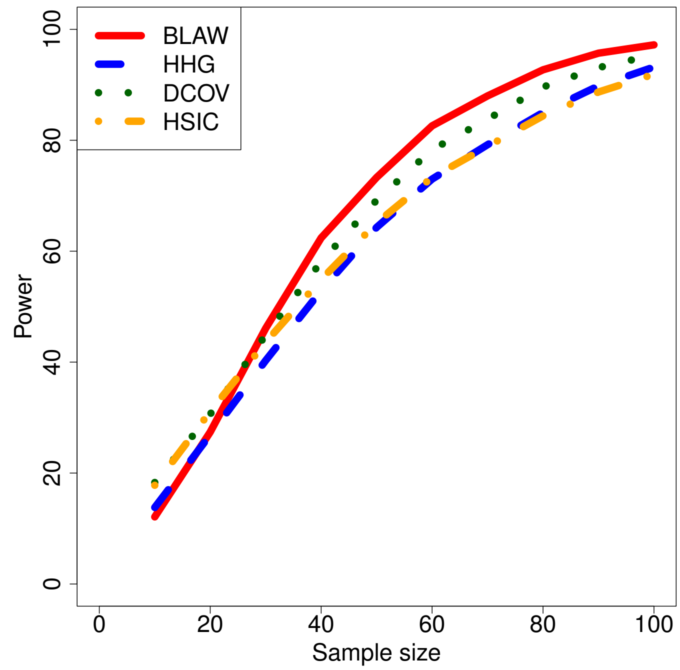

FIG. 3.37. Power in the case of vectors for the Normal ($\rho = 0.3$) copula with Exponential margins

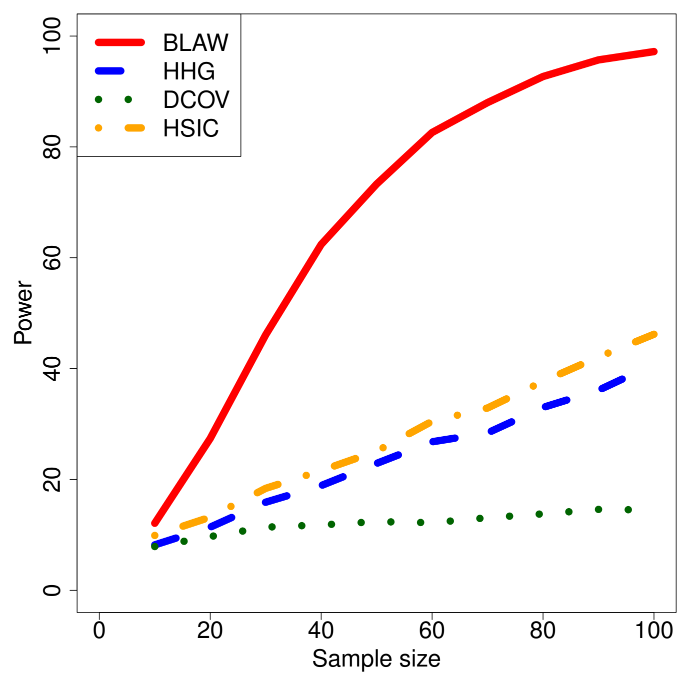

FIG. 3.38. Power in the case of vectors for the Normal ($\rho = 0.3$) copula with Cauchy margins

3.4. Discussion of power results

As a first global comment, let us say that empirically the test we propose "works." Indeed, in all examples when the sample size increases, the power increases, which is what is to be expected. Now we give some specific comments about the examples presented, comparing the power of our proposed test to the power of the other tests for which we did simulations.

In the random variables examples taken from Heller et al. [15] (W-shape, Parabola, Two parabolas, Diamond, Circle), we note that BLAW has a similar or better power then the other tests four times out of five, being worst then HHG and HSIC (but still better then DCOV) only in the case of the 'Diamond' dependence. In the examples that follow (Linear, Exponential, Sine), all four tests have relatively similar power, although we note that DCOV is the winner (by a little) each time, while BLAW is slightly better then HHG each time. In the copula examples, we get more differences between the powers. We note that BLAW is better then HHG eight times out of nine (still being quite similar to HHG in the case of the Gumbel-Exponential example, see Figure 3.20), and is by far the best out of the four tests whenever the marginals are Cauchy. Note that in the copula examples, because BLAW uses ranks, its power is unaltered by the choice of marginals. We argue this is a desirable property for an independence test, as the strength of the dependence is in no way influenced by the choice of the marginals. On the contrary, for the three other tests the power is significantly influenced by the choice of marginals.

In the random vectors examples taken from Heller et al. [15], BLAW is the best test for the 'Log' dependence, although it is worst then HHG in the 'Epsilon' and 'Beta' examples. Then however, in the 'pairwise independence' example BLAW performs way better then HHG, and in the 'big noise' example, while BLAW performs well, all three other tests have almost no power. This is quite interesting, not to say surprising. It means that when testing the independence between two vectors X and Y, if for some reason there is a component in one of the vectors which is not dependent to the other vector and has a 'big' variance, it can compromise greatly the efficiency of HHG, DCOV and HSIC.

For the examples using copulas, conclusions are essentially the same as in the random variables examples. BLAW is better then HHG nine times out of nine, and is drastically better then the three other tests when the marginals are Cauchy.

Hence we are tempted to conclude that BLAW is more robust to the presence of 'extreme' values, as generated for instance when using Cauchy marginals.

Chapter 4

MULTIVARIATE EXTENSION OF HHG

4.1. General idea

As we saw, Heller et al. [15] developed a procedure to test the independence of two random vectors, X and Y. We now propose to use the same technique, but our goal is to test the joint independence between d random vectors, call them $X^{(1)}, X^{(2)}, \cdots, X^{(d)}$, with d being any positive integer. As in the bivariate case, we present first an intuitive idea of this generalization, which we formalize in the following section.

Recall that in the bivariate case, the whole methodology relies on categorizing the data points in 2×2 contingency tables to compute χ^2 statistics. This was the base of the method. Furthermore, recall that, in the case of X and Y being random variables (which is easiest to visualize) we placed rectangles on the scatter plots of the data points. All the sample points could be inside (or outside) the rectangles in their X coordinate, and inside (or outside) the rectangles in their Y coordinate, yielding a 2×2 contingency table, for a total of four categories.

Now with d variables, we can do the same thing. Each point can be classified as inside or outside a d-dimensional zone, according do d different coordinates. Hence, we can categorize each of the data points in a 2^d contingency table. It might help to visualize this with an example where we have three random variables X, Y, Z. The rectangles are now boxes and the scatter plot of observations is now in three dimensions, as shown in Figure 4.1. The points shown on this graph can be classified as being inside or outside the red box according to three coordinates (or axis), x, y and z. The concept of the test stays the same: if there is any choice of box for which knowledge about being inside (or outside) the box according to one coordinate gives some information about being inside (or outside) the box according to another coordinate, then independence between the variables does not hold. Using a large number of 'boxes' defined by the data points will aggregate the evidence against independence. Past this key concept, the rest of the procedure is remarkably similar to the bivariate case:

- Each pair of points (i, j) defines a specific box
- Each box is used to calculate a χ^2 statistic, $S(i, j)$
- We take as our global test statistic the aggregation of all χ^2 statistics:

$$T = \sum_{\substack{i=1 \\ i \neq j}}^{N} \sum_{j=1}^{N} S(i, j)$$

The next section presents the method more formally and establishes a formula for S, which can be implemented in any programming software.

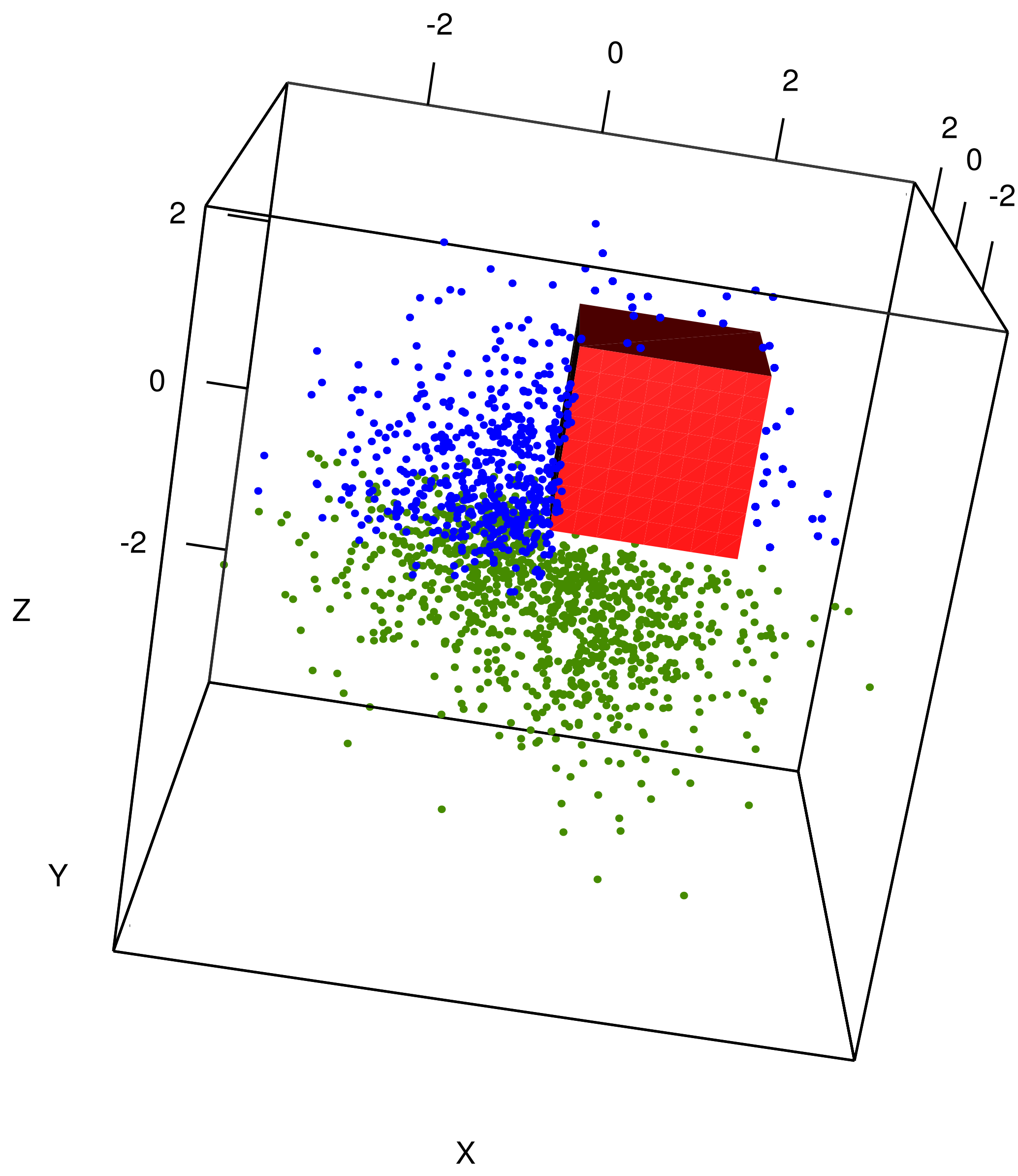

FIG. 4.1. 3-dimensional scatter plot with a box on it

4.2. Test of independence

Let $X^{(1)}, X^{(2)}, \cdots, X^{(d)}$ be d random vectors of any dimension, with respective cumulative distribution functions $F^{(1)}(x_1), F^{(2)}(x_2), \cdots, F^{(d)}(x_d)$. We want to test the hypothesis H_0 of total independence between the d vectors:

$$H_0 : F_{X^{(1)} \ldots X^{(d)}}(x_1, \cdots, x_d) = F_{X^{(1)}}(x_1) \times \cdots \times F_{X^{(d)}}(x_d) \tag{4.2.1}$$

Say we have collected a sample of size N from this distribution. We consider one arbitrary point $(x_i^{(1)}, x_i^{(2)}, \cdots, x_i^{(d)})$ from this sample. This point will be the center of a zone which we will use to categorize all other points in the sample. We use another arbitrary point $(x_j^{(1)}, x_j^{(2)}, \cdots, x_j^{(d)})$ to define d radii $R_0^{(1)} = \text{dist}(x_i^{(1)}, x_j^{(1)}), R_0^{(2)} = \text{dist}(x_i^{(2)}, x_j^{(2)}), \cdots, R_0^{(d)} = \text{dist}(x_i^{(d)}, x_j^{(d)})$. As in the bivariate case, $\text{dist}(\cdot, \cdot)$ is just the euclidean distance.

Now every other point in the sample (there are $N-2$ left) can be categorized as being as close or closer to i then j is, and according to d different coordinates. Consider one specific sample point k (which is not equal to i nor j). For each of the random vectors v we define a categorical function $I_k^{(v)}$ equal to 0 if the distance between the sample point k and the point of dependence is smaller or equal to the radius in the v^{th} component, and equal to 1 if this distance is greater. That is, for $\forall v \in \{1, \cdots, d\}$:

$$I_k^{(v)} = \begin{cases} 0 & \text{if } \text{dist}(x_i^{(v)}, x_k^{(v)}) \leq R_0^{(v)} \\ 1 & \text{if } \text{dist}(x_i^{(v)}, x_k^{(v)}) > R_0^{(v)} \end{cases}.$$

As there are d random vectors, a specific point k can fall into 2^d different categories, i.e. for one given coordinate v the point k can be within the zone defined by the radius $R_0^{(v)}$, or outside the zone. As there are d such coordinates in total, this yields 2^d possibilities. Said otherwise, we categorize each point according to d categories, and for each category there are only two possibilities: being inside or outside the 'zone'. Then the results for the whole sample can be summarized in a 2^d contingency table, or if you prefer a table with

$$\underbrace{2 \times 2 \times \cdots \times 2}_{d \text{ times}} = 2^d$$

cells.

Now, denote by $E(t_1, t_2, \cdots, t_d)$ the expected number of points to fall in a specific cell $(t_1, t_2, \cdots, t_d)$ where $t_v \in \{0, 1\}$, $v \in \{1, \cdots d\}$. Under H_0 we have:

$$\begin{aligned} E(t_1, \cdots, t_d) &= (N-2) \times \mathrm{P}[I_k^{(1)} = t_1, \cdots, I_k^{(d)} = t_d] \\ &= (N-2) \times \prod_{v=1}^{d} \mathrm{P}[I_k^{(v)} = t_v]. \end{aligned}$$

Replacing the theoretical probabilities (which in general we do not know) with the empirical probabilities we get:

$$\begin{aligned} \hat{E}(t_1, \cdots, t_d) &= (N-2) \times \prod_{v=1}^{d} \frac{\sum\limits_{\substack{k=1 \\ k \notin \{i,j\}}}^{N} \mathbb{1}(I_k^{(v)} = t_v)}{N-2} \\ &= \frac{1}{(N-2)^{d-1}} \times \prod_{v=1}^{d} \left(\sum_{\substack{k=1 \\ k \notin \{i,j\}}}^{N} \mathbb{1}(I_k^{(v)} = t_v) \right). \end{aligned}$$

We denote by $A(t_1, t_2, \cdots, t_d)$ the number of sample points observed in cell $(t_1, t_2, \cdots, t_d)$. In other words:

$$A(t_1, \cdots, t_d) = \sum_{\substack{k=1 \\ k \notin \{i,j\}}}^{N} \prod_{v=1}^{d} \mathbb{1}(I_k^{(v)} = t_v).$$

We then write the classic Pearson's test statistic S for a d-dimensional contingency table. Letting $(t_1, \cdots, t_d) = t$ we get:

$$S = \sum_t \frac{[A(t) - \hat{E}(t)]^2}{\hat{E}(t)}.$$

Here we call this statistic S to make the notation less cluttered, but keep in mind it is calculated for one specific pair of points (i, j) hence we could have called it $S(i, j)$. The same goes for the quantities $A(t), E(t)$ and $\hat{E}(t)$. A more precise yet cluttered notation would be $A(i, j)(t), E(i, j)(t)$ and $\hat{E}(i, j)(t)$.

Also for simplification purposes we let $\hat{E}(t) = e(t)/M^{d-1}$, with $M = N - 2$, and rewrite S:

$$\begin{aligned} S =& \sum_t \frac{[A(t) - e(t)/M^{d-1}]^2}{e(t)/M^{d-1}} \\ =& \sum_t \frac{M^{d-1}}{e(t)} \cdot \left(\frac{M^{d-1}A(t) - e(t)}{M^{d-1}} \right)^2 \\ =& \frac{1}{M^{d-1}} \sum_t \frac{1}{e(t)} \cdot \left(M^{2(d-1)}A(t)^2 - 2M^{d-1}A(t)e(t) + e(t)^2 \right) \end{aligned}$$

$$=\frac{1}{M^{d-1}}\sum_t \left(\frac{M^{2(d-1)}A(t)^2}{e(t)} - 2M^{d-1}A(t) + e(t)\right).$$

Which finally yields:

$$S = \sum_t \frac{M^2 A(t)^2}{e(t)} - M \tag{4.2.2}$$

where we used the fact that $\sum_t A(t) = M$ and $\sum_t e(t) = M^d$. This is because $A(t)$ is the number of points in one particular cell t, hence summing the $A(t)$ over all possible cells gives the total number of points categorized, M. The second identity is less obvious, so a proof is given in Appendix A.3.

Note that the statistic S can be calculated only if $e(t)$ is non zero for each of the 2^d possible t. It is set to 0 otherwise. Now, as stated before this gives one statistic, for one particular choice of two sample points (i, j). As in the bivariate case, we calculate the statistic S for each of the $N(N-1)$ pair of points (i, j).

Then finally, our test statistic is the aggregated sum of all $S(i, j)$:

$$T = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N} S(i, j). \tag{4.2.3}$$

Note that T is a sum over $N(N-1)$ terms because there are $N(N-1)$ ways of choosing two points out of N points if the order matters. Indeed, here the order matters: $S(i, j) \neq S(j, i)$.

As in the original bivariate version from Heller et al. [15], we don't know what is null distribution of this test statistic T, and hence to calculate it's p-value we use a permutation method as described previously in section 3.2.

4.3. Proof of the strictly discrete case

For this multivariate extension of HHG, in the case where all the d random vectors are strictly discrete, we prove that the proposed test is consistent whenever H_0 is false. This is of course not the most general case, but still a nice result.

Suppose $X^{(1)} \in \mathbb{R}^{p_1}, \ldots, X^{(d)} \in \mathbb{R}^{p_d}$ are all strictly discrete with a countable support. H_0 is false implies that there exists at least one point $(x_0^{(1)}, \cdots, x_0^{(d)})$ with probability different then 0 such that :

$$\Pr[X^{(1)} = x_0^{(1)}, \ldots, X^{(d)} = x_0^{(d)}] > \Pr[X^{(1)} = x_0^{(1)}] \times \cdots \times \Pr[X^{(d)} = x_0^{(d)}]. \tag{4.3.1}$$

There is also at least one point $(x_0^{(1)*}, \cdots, x_0^{(d)*})$ such that:

$$\Pr[X^{(1)} = x_0^{(1)*}, \ldots, X^{(d)} = x_0^{(d)*}] < \Pr[X^{(1)} = x_0^{(1)*}] \times \cdots \times \Pr[X^{(d)} = x_0^{(d)*}]. \tag{4.3.2}$$

Otherwise, we wouldn't have

$$\sum_{x^{(1)},\ldots,x^{(d)}} \Pr(x^{(1)}, \ldots, x^{(d)}) = 1. \tag{4.3.3}$$

Indeed, imagine this was not the case. That is, imagine we start with a valid distribution that respects H_0. We say valid in the sense that the sum of the probabilities over each mass point equals one. Now we alter this distribution to obtain a distribution under H_1, but we do so by changing the probability assigned to one point and one point only. This yields that, only for this point, (4.3.1) OR (4.3.2) is true. But then, automatically, this new distribution is not valid, i.e. (4.3.3) is false. Hence, we must have that under any valid distribution H_1 at least one point satisfies (4.3.1) AND at least one point satisfies (4.3.2).

From now on, for a point $(x_0^{(1)}, \cdots, x_0^{(d)})$ satisfying (4.3.1), and we know such a point exists under H_1, we denote:

$$\begin{aligned} p_0 &= \Pr[X^{(1)} = x_0^{(1)}, \ldots, X^{(d)} = x_0^{(d)}] \\ p_0^{\perp} &= \Pr[X^{(1)} = x_0^{(1)}] \times \cdots \times \Pr[X^{(d)} = x_0^{(d)}]. \end{aligned}$$

Out of the N points from a sample, we expect $N \cdot p_0$ to have values $(x_0^{(1)}, \ldots, x_0^{(d)})$. Let i_0 and j_0 be two such points. That is:

$$(x_{i_0}^{(1)}, \ldots, x_{i_0}^{(d)}) = (x_{j_0}^{(1)}, \ldots, x_{j_0}^{(d)}) = (x_0^{(1)}, \ldots, x_0^{(d)})$$

Obviously, in each of the d coordinates the distance between the two points i_0 and j_0 is 0. In the notation used previously (section 4.2, p. 76), $R_0^{(1)} = \cdots = R_0^{(d)} = 0$. Hence:

$$A(t_1 = 0, \cdots, t_d = 0)(i_0, j_0) = \sum_{\substack{k=1 \\ k \notin \{i_0, j_0\}}}^{N} \mathbb{1}\{\mathrm{dist}(x_{i_0}^{(1)}, X_k^{(1)}) \leq 0\} \times \cdots \times \mathbb{1}\{\mathrm{dist}(x_{i_0}^{(d)}, X_k^{(d)}) \leq 0\}$$

$$= \sum_{\substack{k=1 \\ k \notin \{i_0, j_0\}}}^{N} \mathbb{1}\{X_k^{(1)} = x_0^{(1)}, \ldots, X_k^{(d)} = x_0^{(d)}\}$$

Denote $A(t_v = 0)(i_0, j_0)$ the number of sample points as close to i_0 than j_0 is in their vth component:

$$A(t_v = 0)(i_0, j_0) = \sum_{\substack{k=1 \\ k \notin \{i_0, j_0\}}}^{N} \mathbb{1}\{\mathrm{dist}(x_{i_0}^{(1)}, X_k^{(1)}) \leq 0\}$$

$$= \sum_{\substack{k=1 \\ k \notin \{i_0, j_0\}}}^{N} \mathbb{1}\{X_k^{(1)} = x_0^{(v)}\}$$

By the law of large numbers, the observed frequencies above converge in probability to the theoretical probabilities:

$$\lim_{N \to \infty} \frac{A(t_1 = 0, \ldots, t_d = 0)(i_0, j_0)}{N - 2} = p_0,$$

$$\lim_{N \to \infty} \frac{A(t_v = 0)(i_0, j_0)}{N - 2} = \Pr[X^{(v)} = x_0^{(v)}], \qquad \forall v \in (1, \ldots, d).$$

Now, recall:

$$S(i_0, j_0) = \sum_t \frac{[A(i_0, j_0)(t) - \hat{E}(i_0, j_0)(t)]^2}{\hat{E}(i_0, j_0)(t)}.$$

It is enough to look at the first term of this sum, with $t = (t_1 = 0, \cdots, t_d = 0)$, i.e. the term:

$$S_1(i_0, j_0) = \frac{\left\{A(t_1 = 0, \cdots, t_d = 0)(i_0, j_0) - \frac{\prod_{v=1}^{d} A(t_v=0)(i_0,j_0)}{(N-2)^{d-1}}\right\}^2}{\frac{\prod_{v=1}^{d} A(t_v=0)(i_0,j_0)}{(N-2)^{d-1}}}.$$

Now, because the points i_0, j_0 violate independence, with $N \to \infty$ the ratio $S_1(i_0, j_0)/(N - 2)$ won't go to zero. That is, almost surely:

$$\lim_{N \to \infty} \frac{S_1(i_0, j_0)}{N - 2} = \lim_{N \to \infty} \frac{1}{N - 2} \frac{\{A(t_1 = 0, \cdots, t_d = 0)(i_0, j_0) - \frac{\prod_{v=1}^{d} A(t_v=0)(i_0,j_0)}{(N-2)^{d-1}}\}^2}{\frac{\prod_{v=1}^{d} A(t_v=0)(i_0,j_0)}{(N-2)^{d-1}}}$$

$$= \lim_{N\to\infty} \frac{\frac{1}{(N-2)^2}\{A(t_1=0,\cdots,t_d=0)(i_0,j_0) - \frac{\prod_{v=1}^d A(t_v=0)(i_0,j_0)}{(N-2)^{d-1}}\}^2}{\frac{\prod_{v=1}^d A(t_v=0)(i_0,j_0)}{(N-2)^d}}$$

$$= \frac{\lim_{N\to\infty}\{\frac{A(t_1=0,\cdots,t_d=0)(i_0,j_0)}{N-2} - \frac{\prod_{v=1}^d A(t_v=0)(i_0,j_0)}{(N-2)^d}\}^2}{\lim_{N\to\infty} \frac{\prod_{v=1}^d A(t_v=0)(i_0,j_0)}{(N-2)^d}}$$

$$= \frac{\{p_0 - p_0^\perp\}^2}{p_0^\perp}$$

where we used Slutzky's theorem (Serfling [31], p. 19), to replace the empirical probabilities in $S_1(i_0, j_0)$ by theoretical probabilities.

Because $S_1(i_0, j_0)/(N-2)$ converges almost surely to a positive constant $c' = \frac{\{p_0 - p_0^\perp\}^2}{p_0^\perp} > 0$, we have:

$$S_1(i_0, j_0) > (N-2)c'/2 \tag{4.3.4}$$

with probability going to 1 as $N \to \infty$.

Now, for any given pair of points (k, l), the probability that both points are equal to $(x_0^{(1)}, \cdots, x_0^{(d)})$ is p_0^2. Out of the $N(N-1)$ pairs of points from the sample, we let N_0^* be the number of pairs such as the pair (i_0, j_0), that is pairs with both points equal to $(x_0^{(1)}, \cdots, x_0^{(d)})$. The counting variable N_0^* can be written as a sum of (dependent) Bernoulli:

$$N_0^* = \sum_{\substack{k=1 \\ k\neq l}}^N \sum_{l=1}^N B_{kl}, \qquad B_{kl} \sim Bernoulli(p_0^2)$$

where $B_{kl} = 1$ if both points k and l equal $(x_0^{(1)}, \cdots, x_0^{(d)})$, and 0 otherwise. Hence, $E[N_0^*] = N(N-1)p_0^2$. We prove below that for $N \to \infty$:

$$\Pr[N_0^* > \frac{N(N-1)p_0^2}{2}] = 1. \tag{4.3.5}$$

Now recall the definition of our global test statistic T:

$$T = \sum_{\substack{i=1 \\ i\neq j}}^N \sum_{j=1}^N S(i,j).$$

Because for any two sample points i and j we have $S(i,j) \geq S_1(i,j)$, and because our test statistic T is the sum over all $S(i,j)$ for every possible pairs of points, we have the following lower bound for T:

$$T \geq \sum_{\substack{i=1 \\ i \neq j}}^{N} \sum_{j=1}^{N} S_1(i,j).$$

From 4.3.4 and 4.3.5 it follows that, for $N \to \infty$:

$$T \geq \sum_{\substack{i=1 \\ i \neq j}}^{N} \sum_{j=1}^{N} S_1(i,j) > \frac{1}{2} N(N-1) p_0^2 \cdot \frac{(N-2)c'}{2} \geq \delta N^3$$

for a positive constant $\delta > 0$, with probability 1.

To complete the proof we invoke the same argument as in the last paragraph of the Appendix in Heller et al. [15]. That is, under the null hypothesis H_0, $S(i,j)$ is asymptotically χ^2, so the expectation of T under the null is of order of magnitude $N(N-1)$. But since for any H_1 we just established that T has order of magnitude at least N^3, it follows that asymptotically $T|H_1$ is always bigger than $T|H_0$, hence H_0 is rejected with probability 1.

This completes the proof, although we still need to prove statement (4.3.5). To do so we need the variance of N_0^*. Then, because the variance is of magnitude N^3, and the mean of magnitude N^2, we will use Chebyshev inequality to establish the result. Recall:

$$N_0^* = \sum_{\substack{k=1 \\ k \neq l}}^{N} \sum_{l=1}^{N} B_{kl}.$$

This is a sum of $N(N-1)$ elements. We look at the covariance of two of those elements:

$$\mathrm{COV}[B_{kl}; B_{mn}].$$

There are three possible cases (i.e. yielding different covariances):

(1) If k, l, m, n are all different, then:

$$\begin{aligned} \mathrm{COV}[B_{kl}; B_{mn}] &= \mathrm{E}[B_{kl} \cdot B_{mn}] - \mathrm{E}[B_{kl}] \cdot \mathrm{E}[B_{mn}] \\ &= \Pr[(x_k, y_k) = (x_l, y_l) = (x_m, y_m) = (x_n, y_n) = (x_0, y_0)] - p_0^4 \\ &= p_0^4 - p_0^4 = 0. \end{aligned}$$

(2) If out of k, l, m, n exactly two are the same, say *WLOG* $k = m$:

$$\mathrm{COV}[B_{kl}; B_{kn}] = \mathrm{E}[B_{kl} \cdot B_{kn}] - \mathrm{E}[B_{kl}] \cdot \mathrm{E}[B_{kn}]$$

$$
\begin{aligned}
&= \Pr[(x_k, y_k) = (x_l, y_l) = (x_n, y_n) = (x_0, y_0)] - p_0^4 \\
&= p_0^3 - p_0^4 = p_0^3(1 - p_0)
\end{aligned}
$$

(3) Two pairs of elements are equal, in other words $m = l$ and $n = k$:

$$
\begin{aligned}
\mathrm{COV}[B_{kl}; B_{lk}] &= \mathrm{E}[B_{kl} \cdot B_{lk}] - \mathrm{E}[B_{kl}] \cdot \mathrm{E}[B_{lk}] \\
&= \Pr[(x_k, y_k) = (x_l, y_l) = (x_0, y_0)] - p_0^4 \\
&= p_0^2 - p_0^4 = p_0^2(1 - p_0^2)
\end{aligned}
$$

We can now calculate the variance of N_0^*:

$$
\begin{aligned}
\mathrm{VAR}[N_0^*] &= \mathrm{VAR}\left[\sum_{\substack{k=1 \\ k \neq l}}^{N} \sum_{l=1}^{N} B_{kl}\right] \\
&= \sum_{\substack{k=1 \\ k \neq l}}^{N} \sum_{l=1}^{N} \mathrm{VAR}[B_{kl}] + \sum_{\tau \neq \alpha} \mathrm{COV}[B_\tau, B_\alpha] + \sum_{\gamma \neq \theta} \mathrm{COV}[B_\gamma, B_\theta]
\end{aligned}
$$

where τ and α index the pairs of case 2, and γ and θ index the pairs of case 3. There are:

- $2\,[N(N-1) \times 4(N-2)]$ terms of case 2
- $2\,[N(N-1)]$ terms of case 3

Hence:

$$
\begin{aligned}
\mathrm{VAR}[N_0^*] =& N(N-1)p_0^2(1-p_0^2) + \\
& 2N(N-1)4(N-2)p_0^3(1-p_0) + \\
& 2N(N-1)p_0^2(1-p_0^2) \\
=& N(N-1)p_0^2 \cdot [(1-p_0^2) + 8(N-2)p_0(1-p_0) + 2(1-p_0^2)].
\end{aligned}
$$

Therefore, there is a constant $\xi > 0$ such that $\mathrm{VAR}[N_0^*] \leq \xi N^3$. We then use Chebyshev inequality, for any positive t:

$$
\begin{aligned}
\Pr[|N_0^* - E[N_0^*]| \geq t] &\leq \frac{\mathrm{VAR}[N_0^*]}{t^2} \\
\Pr[N_0^* - E[N_0^*] \leq -t] &\leq \frac{\mathrm{VAR}[N_0^*]}{t^2}.
\end{aligned}
$$

Letting $t = E[N_0^*]/2$ we have:

$$
\begin{aligned}
\Pr[N_0^* \leq E[N_0^*]/2] &\leq \frac{\mathrm{VAR}[N_0^*]}{E[N_0^*]^2/4} \\
&\leq \frac{\xi N^3}{p_0^2 N^2 (N-1)^2/4}.
\end{aligned}
$$

This upper bound going to 0 for $\lim N \to \infty$ we have proved the result.

4.4. Power simulations

As we did for the bivariate version of the test, we now investigate the power of our multivariate extension of Heller et al. [15]. We test the independence between three random quantities X, Y and Z. Power simulations are conducted using the steps described in section 3.3.1.

In the bivariate case we saw that in the majority of examples investigated, using the ranks of the observations yielded a better power. We want to see if the same conclusion holds in the multivariate case. Hence here we compute two versions of the test: the 'straight-forward' extension (using the original data) which we label mHHG, and the multivariate version using ranks of the observations, which we label mBLAW.

In subsection 4.4.1 we explore examples where X, Y and Z are random variables (each one having one component). In those cases, we also compare mBLAW and mHHG to the independence tests found in Beran et al. [3], which we label BBL, to the multivariate HSIC from Pfister et al. [23] which we label mHSIC, and finally to the test from Genest and Rémillard [10], which we label GR.

Next, in subsection 4.4.2 we explore examples where X, Y and Z are vectors, both of dimension two: $X = (X_1, X_2)$, $Y = (Y_1, Y_2)$, $Z = (Z_1, Z_2)$. In that case, we only compare mHHG and mBLAW to mHSIC. This is because GR does not apply to random vectors. Note that we took awareness very late that a generalization of GR exists in Kojadinovic and Holmes [19]. Hence we applied this generalization only on one example, see Figure 4.15. Also note that BBL applies to vectors, but as it takes significantly more time to compute in the case of vectors we also only used it in one example, see Figure 4.15.

4.4.1. Random variables examples

The dependence models for the variables X, Y, Z are now presented, and the graphs of power for each example follow thereafter.

- 3D Pairwise independence:

$$X, Y \text{ and } Z_0 \sim N(0,1)$$
$$Z = |Z_0| \cdot \mathrm{sign}(X \cdot Y)$$

Note this is the motivational example from section 1.3.2, but here we are able to test the mutual independence of the triplet X, Y, Z, unlike in the example '2D pairwsie independence' where we had to combine Y and Z has one vector, to test the pairwise independence of X and (Y, Z).

- Cos-Sin dependence:

$$X \text{ and } Y \sim N(0,3)$$
$$Z = \cos(X) + \sin(Y) + N(0,1)$$

- Cos-Exp dependence:

$$X \text{ and } Y \sim N(0,3)$$
$$Z = \cos(X) + \exp(Y/5) + N(0,1)$$

- 3D linear dependence:

$$X \text{ and } Y \sim N(0,1)$$
$$Z = X + Y + N(0,3)$$

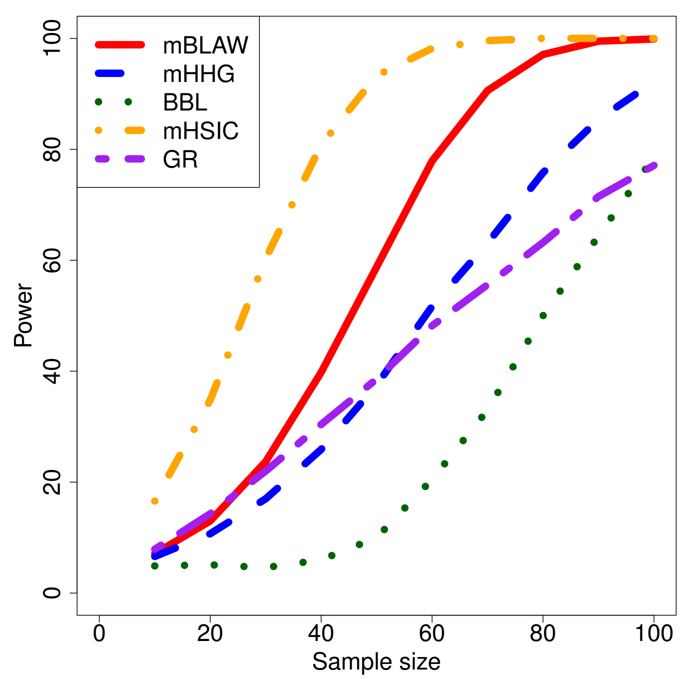

FIG. 4.2. Power in the case of 3D pairwise independent Normals

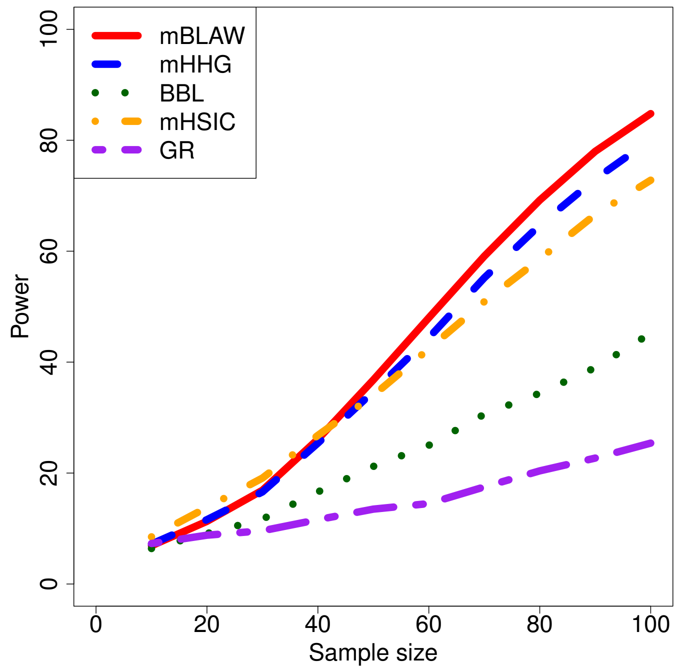

Fig. 4.3. Power in the case of the Cos-Sin dependence

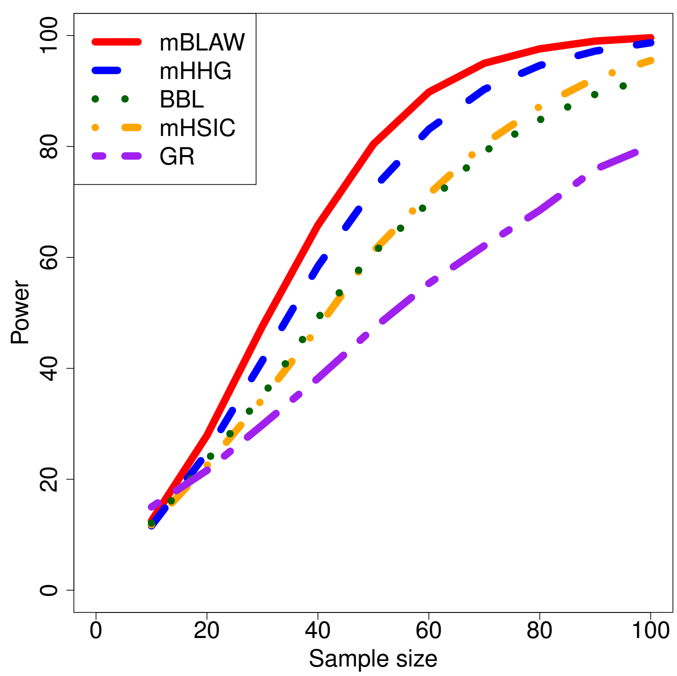

Fig. 4.4. Power in the case of the Cos-Exp dependence

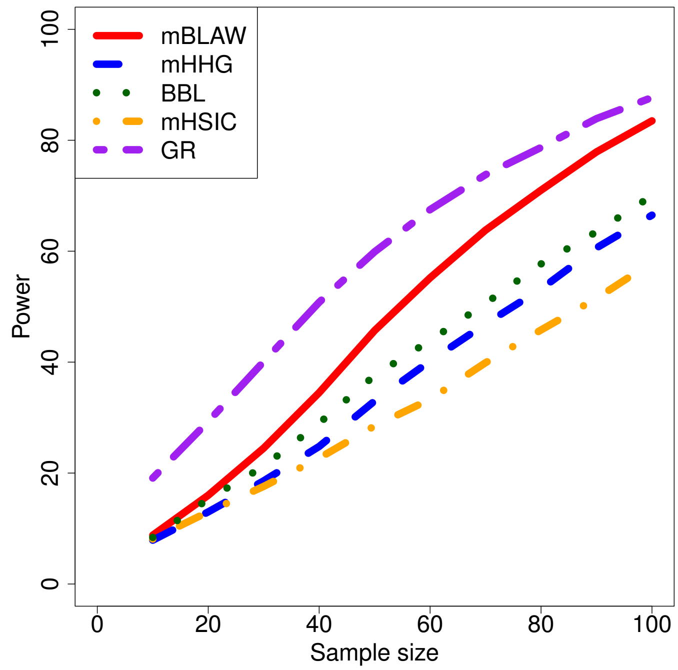

Fig. 4.5. Power in the case of the 3D linear dependence

Next, we again present examples where the data is dependent via copula structures. As in the bivariate examples, three copulas are considered, still Clayton, Gumbel and Normal, with three different marginals: Normal$(0, 1)$, Exponential (scale $= 1$) and Cauchy (scale $= 1$).

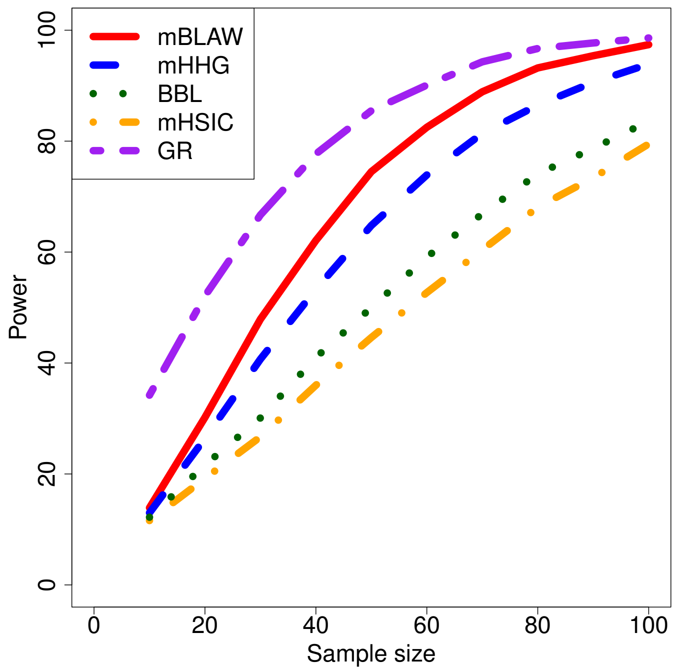

FIG. 4.6. Power in the case of the 3D Clayton copula (0.5) with Normal margins

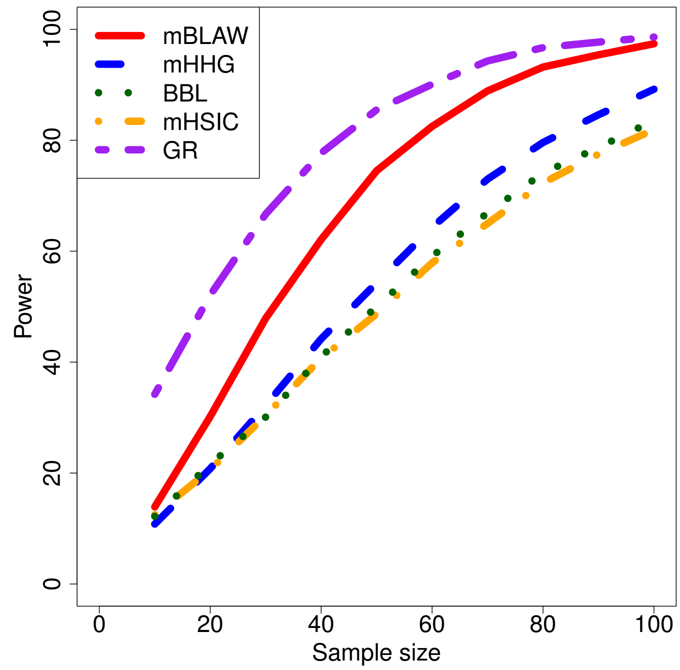

FIG. 4.7. Power in the case of the 3D Clayton copula (0.5) with Exponential margins

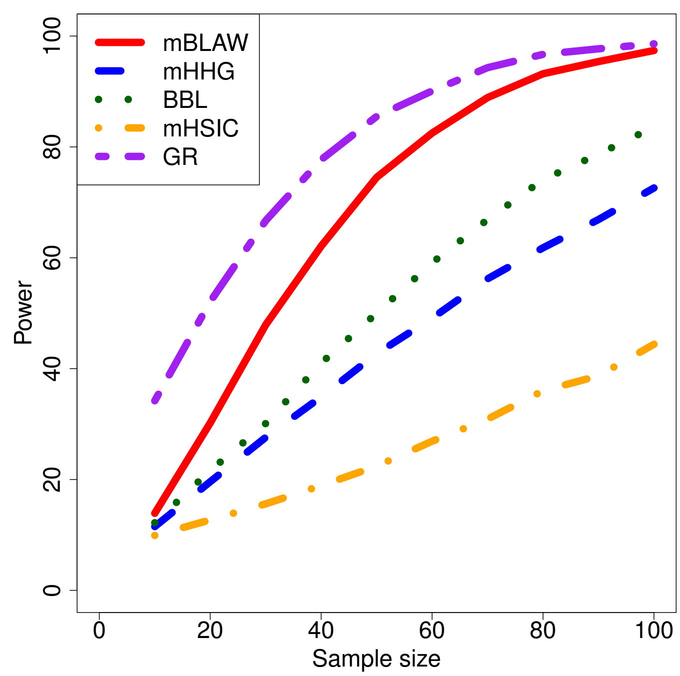

FIG. 4.8. Power in the case of the 3D Clayton copula (0.5) with Cauchy margins

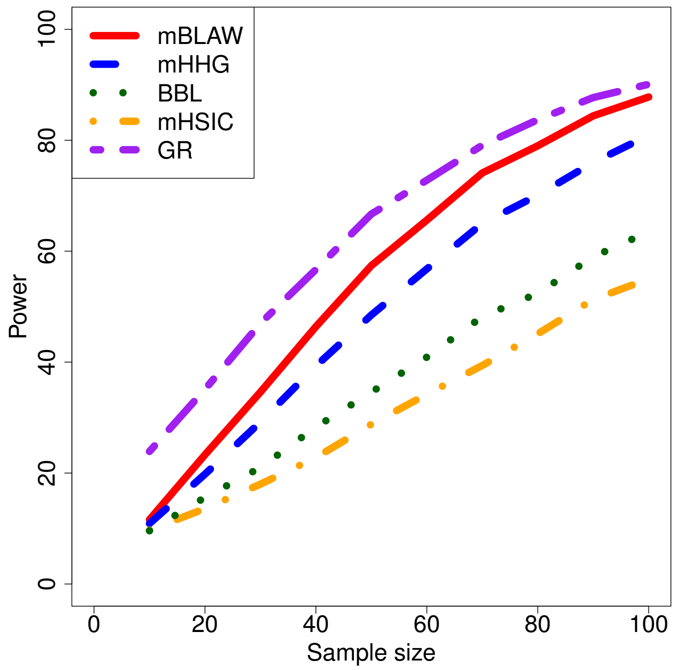

FIG. 4.9. Power in the case of the 3D Gumbel copula (1.2) with Normal margins

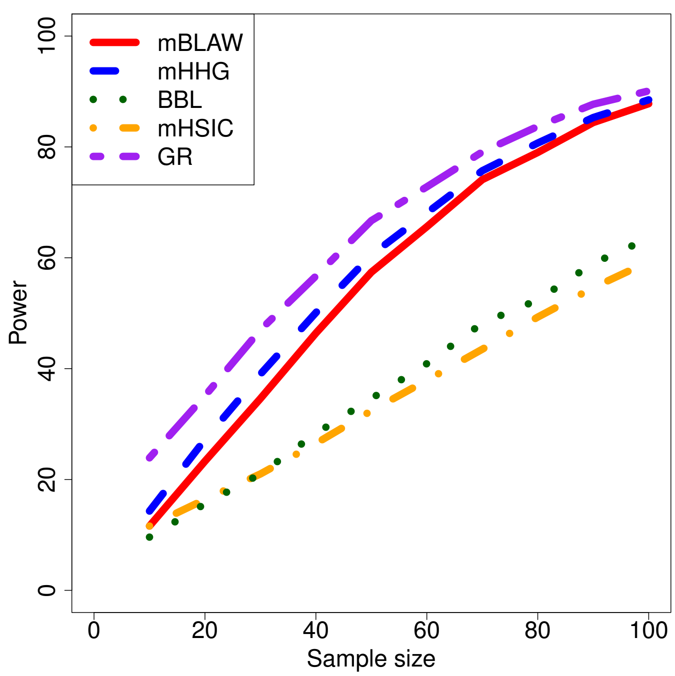

FIG. 4.10. Power in the case of the 3D Gumbel copula (1.2) with Exponential margins

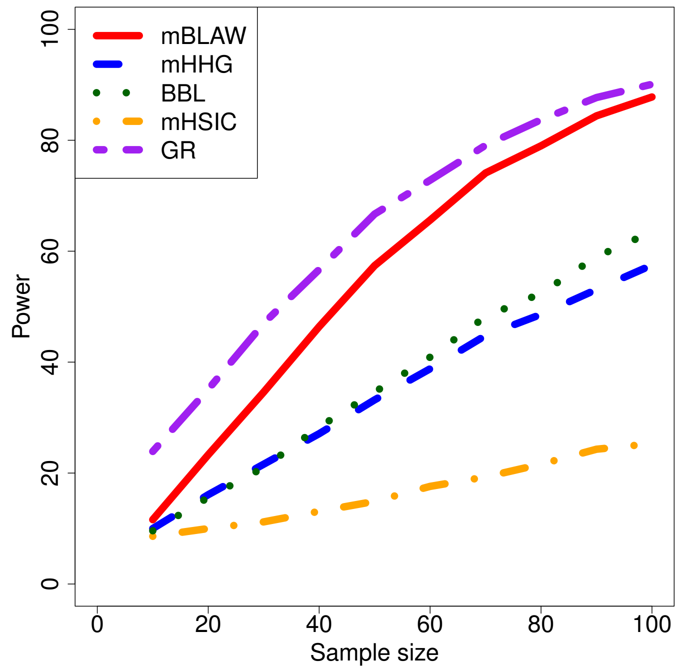

FIG. 4.11. Power in the case of the 3D Gumbel copula (1.2) with Cauchy margins

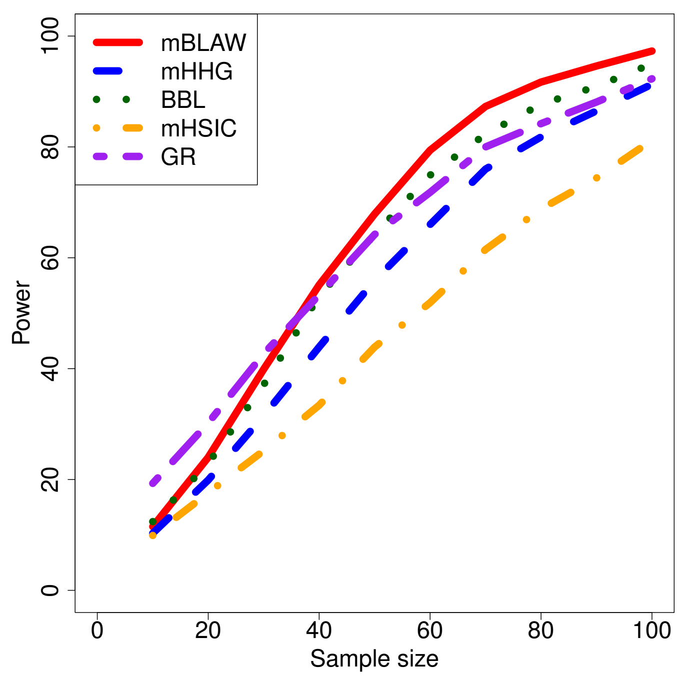

FIG. 4.12. Power in the case of the 3D Normal copula ($\rho_{x,y} = 0.0, \rho_{x,z} = 0.1, \rho_{y,z} = 0.5$) with Normal margins

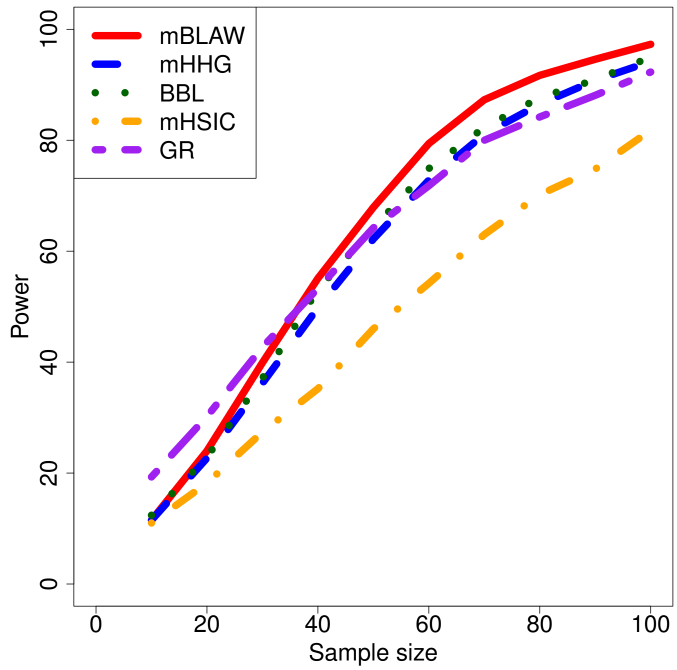

FIG. 4.13. Power in the case of the 3D Normal copula ($\rho_{x,y} = 0.0, \rho_{x,z} = 0.1, \rho_{y,z} = 0.5$) with Exponential margins

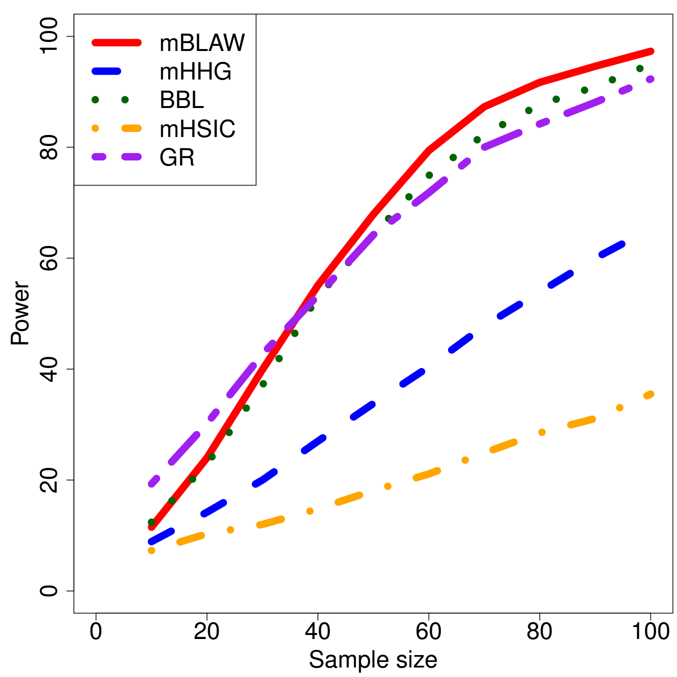

FIG. 4.14. Power in the case of the 3D Normal copula ($\rho_{x,y} = 0.0, \rho_{x,z} = 0.1, \rho_{y,z} = 0.5$) with Cauchy margins

4.4.2. Random vectors examples

Here we consider examples where X, Y and Z are vectors, all in two dimensions: $X = (X_1, X_2)$, $Y = (Y_1, Y_2)$ and $Z = (Z_1, Z_2)$. The first two examples we present are derived from the motivational example in section 1.3.2. Note that we computed, only in the first example, the powers for two additional tests: Beran et al. [3] (labeled BBL) and Kojadinovic and Holmes [19] (labeled KOJA). Because BBL is slow to compute on random vectors, for this first example we used sample sizes $N = \{10, 20, \dots 70\}$ and $B = 5000, M = 10000$. All other examples are computed with $B = 10000, M = 50000$.

- 3D vectors with pairwise independence: case "mixed"

$$
\begin{aligned}
&X_1, X_2, Y_0, Y_1, Z_0, Z_1 \sim N(0,1)\\
&Y_2 = |Y_0| \cdot \operatorname{sign}(X_1 \cdot Z_1)\\
&Z_2 = |Z_0| \cdot \operatorname{sign}(X_2 \cdot Y_1)
\end{aligned}
$$

  We label this example "mixed" because only joint information about X and Z gives information about Y_1. Likewise, joint information about X and Y gives information about Z_2.

- 3D vectors with pairwise independence: case "hidden"

$$
\begin{aligned}
&X_1, X_2, Y_1, Z_0, Z_1, Z_2 \sim N(0,1)\\
&Y_2 = |Z_0| \cdot \operatorname{sign}(X_1 \cdot X_2)
\end{aligned}
$$

  We label this example "hidden" because here the dependence is "hidden" in some sense, harder to find. This is so because Z is completely independent of both X and Y, while only Y_2 is dependent of X.

Next, and without much surprise, we present example where the dependent data is generated using three different copulas (Clayton, Gumbel, Normal) and three marginals (Normal, Exponential, Cauchy).

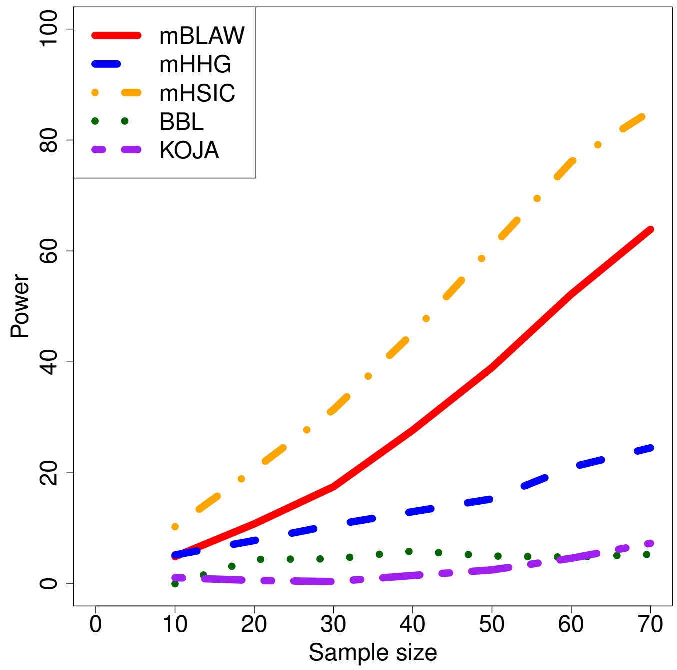

FIG. 4.15. Power in the case of vectors with pairwise independent components: case "mixed"

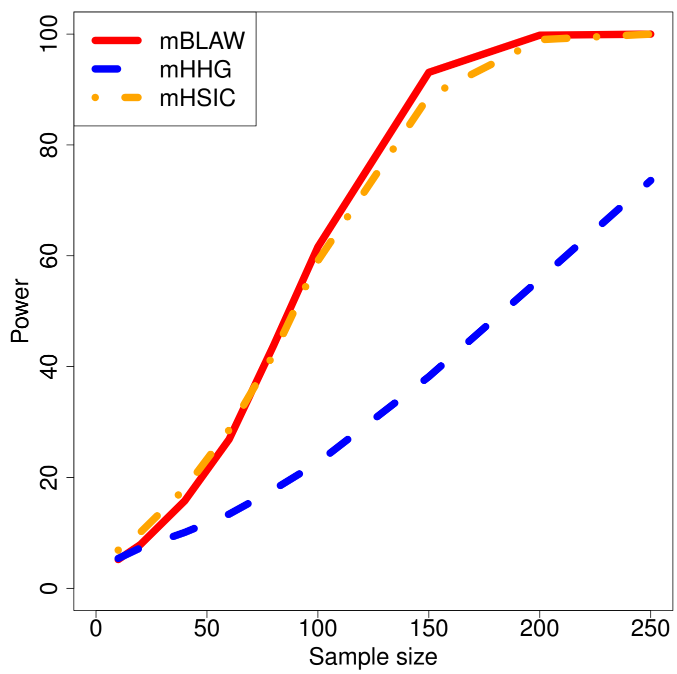

FIG. 4.16. Power in the case of vectors with pairwise independent components: case "hidden"

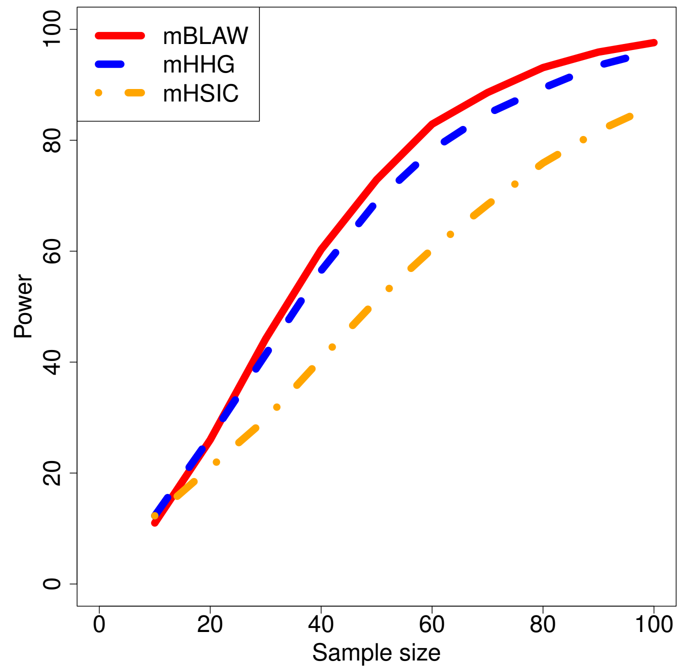

FIG. 4.17. Power in the case of the 3D Clayton copula (0.3) with Normal margins for vectors

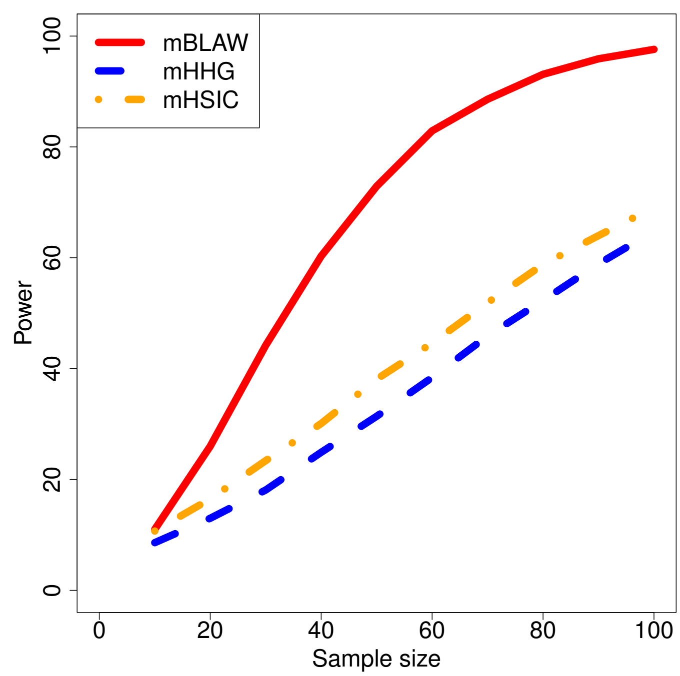

FIG. 4.18. Power in the case of vectors for the 3D Clayton copula (0.3) with Exponential margins

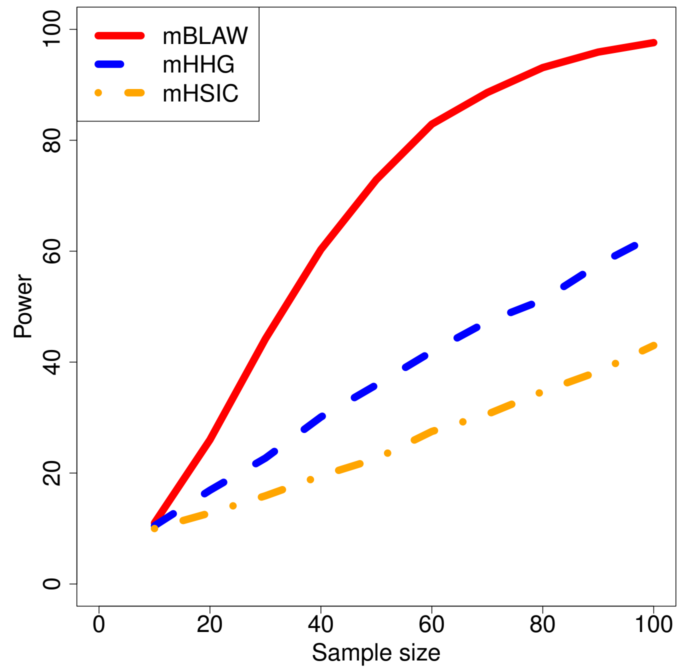

FIG. 4.19. Power in the case of vectors for the Clayton copula (0.3) with Cauchy margins

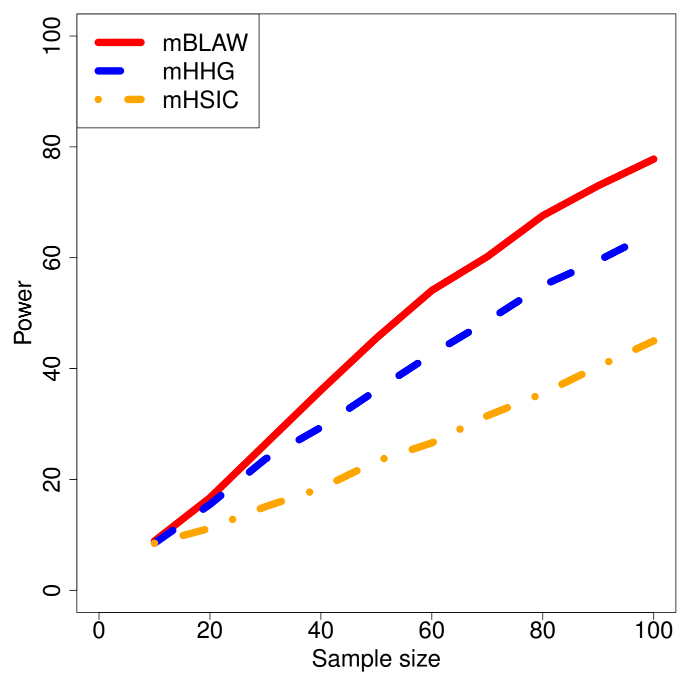

FIG. 4.20. Power in the case of vectors for the Gumbel copula (1.1) with Normal margins

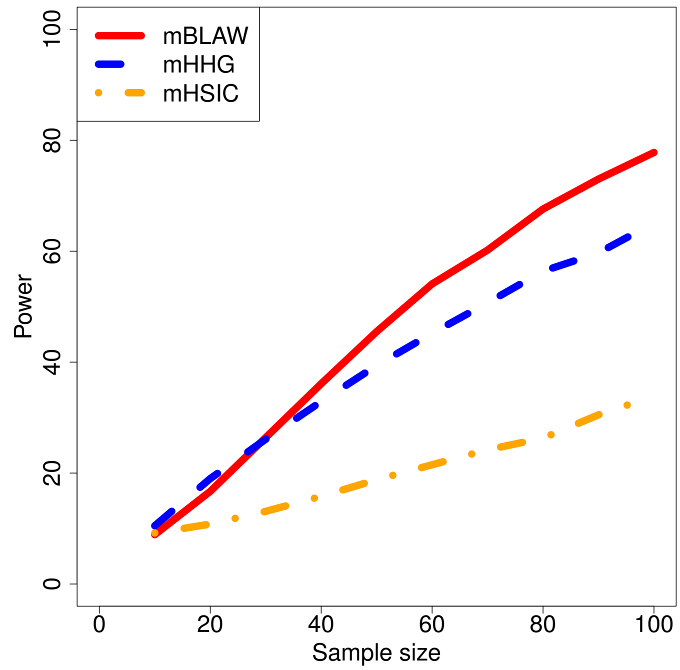

Fig. 4.21. Power in the case of vectors for the Gumbel copula (1.1) with Exponential margins

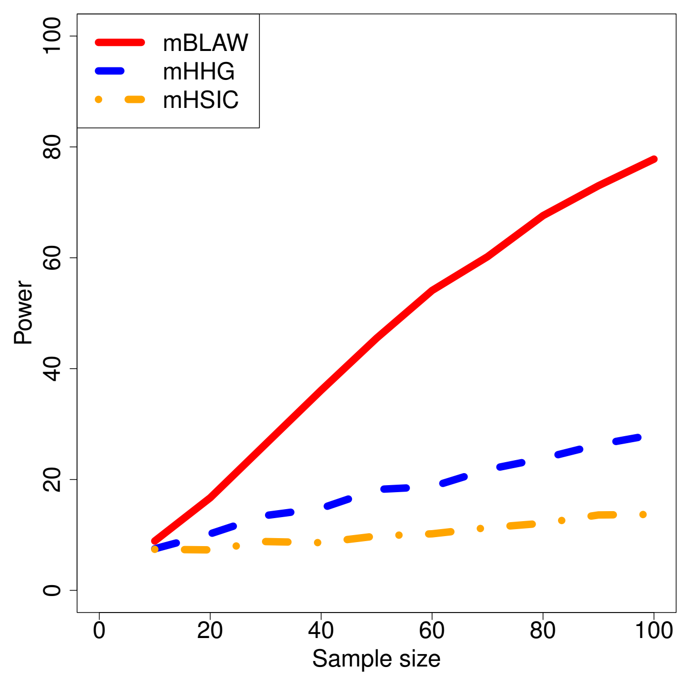

Fig. 4.22. Power in the case of vectors for the Gumbel copula (1.1) with Cauchy margins

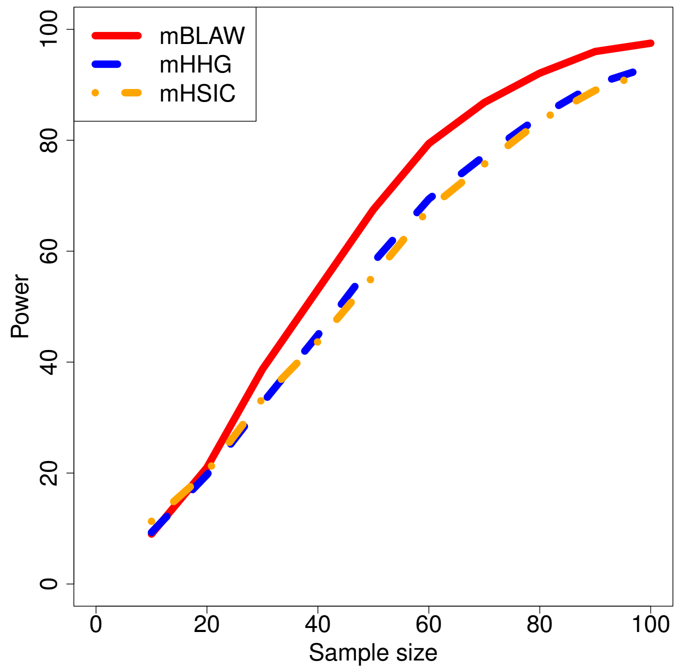

FIG. 4.23. Power in the case of vectors for the Normal copula ($\rho_{x,y} = \rho_{x,z} = 0.1, \rho_{y,z} = 0.3$) with Normal margins

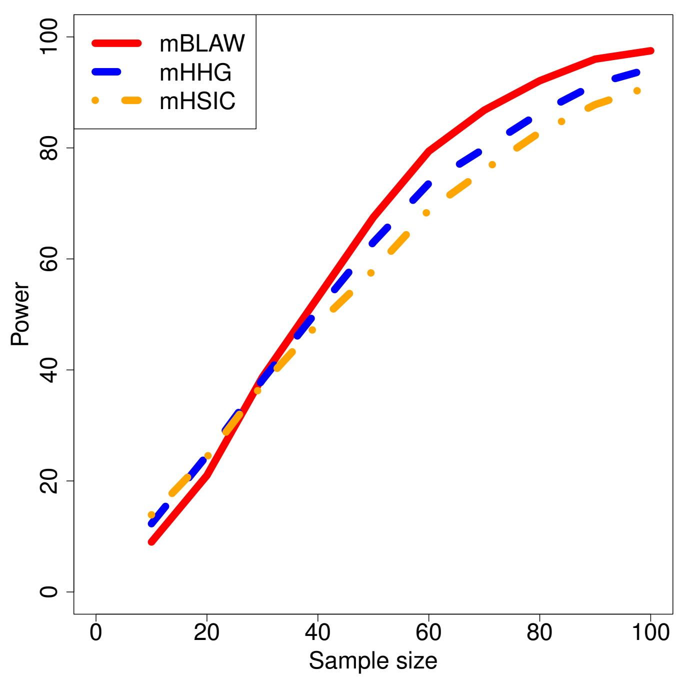

FIG. 4.24. Power in the case of vectors for the Normal copula ($\rho_{x,y} = \rho_{x,z} = 0.1, \rho_{y,z} = 0.3$) with Exponential margins

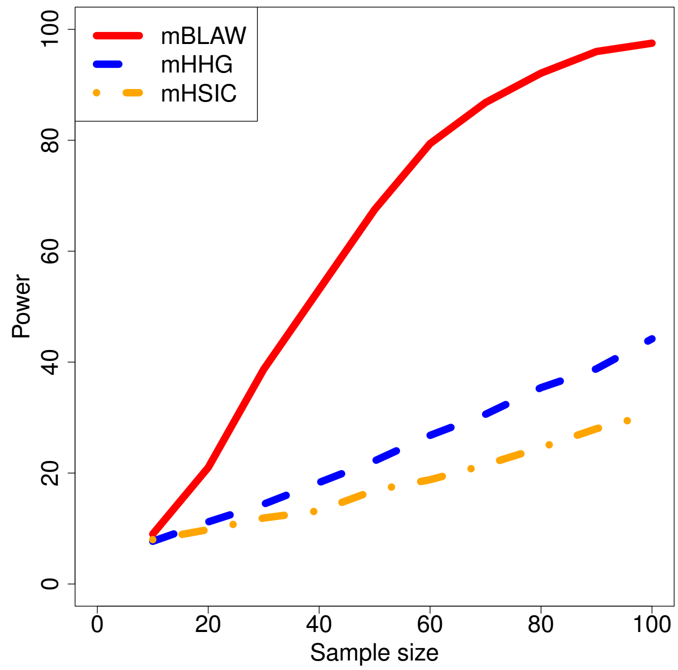

FIG. 4.25. Power in the case of vectors for the Normal copula ($\rho_{x,y} = \rho_{x,z} = 0.1, \rho_{y,z} = 0.3$) with Cauchy margins

4.5. Discussion of power results

Here we start by saying that both the tests we call “mHHG” and “mBLAW” are new, in the sense that mHHG is the extension of Heller et al. [15] presented in section 4, while mBLAW is that same extension, but using ranked data instead of the original data. Although we state our preference for mBLAW, we investigated the power of both tests in comparison to other tests labeled BBL (from Beran et al. [3]), mHSIC (from Pfister et al. [23]) and GR (from Genest and Rémillard [10]).

In the random variables examples, we started with the motivational example from section 1.3.2, where the variables X, Y and Z are pairwise independent but not mutually independent. mBLAW is outperformed by mHSIC, but is still significantly better then mHHG, BBL and GR. Next on the ‘Cos-Sin’ example as well as the ‘Cos-Exp’ example mBLAW is the best test, and GR is the worst. In the ‘linear’ dependence however GR is best, while mBLAW is second best. In the copula examples, six times out of nine (for the Clayton and Gumbel copulas) GR is the best test, with BLAW being the second best. For the Normal copula and sample sizes bigger then 40, mBLAW is the best test.

Perhaps it is not surprising that GR performs well when data is generated via copulas, because it uses the empirical copula processes as base for its methodology. It is also worthy to note that while for obvious reasons the power of GR and BLAW is unaltered by the marginals of X, Y and Z, it is also the case for the BBL test. However, as in the bivariate examples, the power of mHHG and mHSIC is greatly influenced by the distribution of the marginals. Again, we consider this to be a flaw, as the choice of marginals does not influence the strength of the dependence.

In the random vectors examples, first we presented two new versions of the “pairwise independence” motivational example. In the first of such examples, which we called “case mixed”, see Figure 4.15, mHSIC performs best and mBLAW is the second best. BBL and KOJA have almost no power, while mHHG has little power.

In the second of those examples (“case hidden”), see Figure 4.16, note that we computed the power for sample sizes from 10 to 250. This is because the dependence here is harder to find, with Z being independent of both X and Y. We see that mBLAW performs way better then mHHG, and is very similar to mHSIC.

Next in the copula examples, the same conclusions as in the random variables examples can be drawn: mBLAW is always better than mHHG, and substantially so when the marginals are Cauchy. It is also always better than mHSIC. Again we see that the power of mHSIC and mHHG varies a lot when the marginals change. This is not the case of mBLAW.

Chapter 5

CONCLUSION

5.1. Motivation for this research

How to test for independence is a fundamental question in statistics. The tools available to answer this question have greatly evolved in the past century. Since the famous Pearson's correlation coefficient was established, increasingly sophisticated tests have been developed. Statisticians working in this field aim at creating methods as general as possible, with good power to detect any form of association between any number of random vectors.

Specifically to test the independence between two random quantities, some tests recently developed have been found to be universally consistent. Perhaps the most popular among them is the distance covariance test from Székely et al. [34]. However, if a test aims at being applicable in the most general of contexts, it is unlikely to be the most powerful in all situations. For instance, Heller et al. [15] established that their test outperforms distance covariance in many examples. Likewise, in the present thesis we saw some examples where both HHG and DCOV were outperformed by HSIC, and vice versa.

Now, as we mentioned before, both the HHG and DCOV tests detect dependence between two random vectors X and Y. This could be seen as a limitation because pairwise independence between all vectors of a set does not imply mutual independence of those vectors. That said, few consistent methods are currently available to test the mutual independence between any number of random vectors. This absence, combined with our belief that HHG had the potential to be further developed, were the main motivations for this thesis. In the end we obtained a non-parametric test of independence between an arbitrary number of random vectors, with arbitrary (possibly different) dimensions.

5.2. Contributions

5.2.1. On bivariate ranked data

In this research we extended the methodology from Heller et al. [15] in two manners. First, we argued that applying their test not on the collected data itself but on the ranks of the collected data would produce a more robust and powerful method. The motivation behind this was explained in section 3.1. To recap, the rank transformation puts all observations on the same scale. This is suitable because the marginal distribution of a random variable has nothing to do with its dependence to other variables. Said otherwise, using the rank transformation strips away the impact of the marginal distributions, to leave only the dependency structure which is what we seek to detect. The bivariate test in Heller et al. [15] applied to ranked-data was labeled the “BLAW” test. We investigated its power on numerous and varied simulated data sets. We compared it to the power of three recent, state-of-the-art independence tests, namely the original HHG, the DCOV and the HSIC tests.

We found that:

- BLAW is consistent to detect all the forms of dependence we investigated. That is, its power always gets bigger if the sample size N gets bigger.
- BLAW has similar or better power then HHG, DCOV and HSIC in most examples. To be more precise, only in examples labeled ‘epsilon dependance’, see Figure 3.26 and ‘quadratic dependence’, see Figure 3.27 did HHG beat BLAW by a notable margin. In the copula examples, both when X and Y were variables and vectors, BLAW was clearly superior to HHG.
- Furthermore, in the copula examples we saw that BLAW was the only test not affected by the choice of marginal distributions. This is a good property for an independence test.
- In particular, we saw that with Cauchy marginals, HHG, DCOV and HSIC performed rather poorly. This means that BLAW is more robust than the other tests in the presence of extreme values.
- In one example in particular labeled ‘big noise’, see Figure 3.29, all tests except BLAW had negligible power. This was a rather surprising result to us. Indeed, while Y was strongly dependent of $X = (X_1, X_2)$ via X_1 (but not via X_2) the fact that X_2 had a ‘big’ variance made those tests practically incapable of detecting the dependence between X and Y.

To summarize, although using ranks is a very simple modification to the original test, we believe the results obtained make it a valuable contribution, as we consistently obtained better power than three state-of-the-art tests.

5.2.2. On multivariate data

Then, we also made the method in Heller et al. [15] more general by extending it to test the mutual independence between any number of random vectors. We realized that the key aspect of their method (i.e. the categorization of all sample points in 2×2 contingency tables to perform multiple χ^2 tests of independence) was perfectly valid in a multivariate context. Indeed, if for two random vectors we could define zones according to two coordinates, then for d random vectors we could define zones according to d coordinates. Then, we could classify each sample point as inside or outside the zones, according two d coordinates. I.e. we could categorize our points in a 2^d contingency table. In turn, this table could be used to calculate a χ^2 statistic just as in the bivariate case. In total, $N(N-1)$ such χ^2 statistics were calculated and their sum yielded our global test statistic T. Formalization of this procedure was done in section 4.2.

We investigated the power of this new test on various examples. Because using ranks worked well in the bivariate case we tried it in the multivariate case as well. The direct multivariate extension of HHG was labeled "mHHG", while the version using ranks was labeled "mBLAW". In examples involving random variables, we compared mHHG and mBLAW to three other tests, namely mHSIC, BBL and GR. In random vectors examples we compared them only to mHSIC.

We found that:

- Akin the bivariate case, mBLAW is consistent to detect all the forms of dependence we investigated. That is, its power always gets bigger if the sample size N gets bigger.
- In the random variables examples using copulas, GR performed better than mBLAW six times out of nine, but mBLAW was still the second best test in each of those six examples, and also perfomed best in the three remaining copula examples. Then, GR performed rather poorly in the 'pairwise independence', 'Cos-Sin' and 'Cos-Exp' examples, see Figures 4.2 to 4.4, while mBLAW did good, being beaten by mHSIC only in the 'pairwise independence' case.

- Akin to the bivariate case, mBLAW was unaffected by the choice of marginals, unlike mHHG and mHSIC that were strongly affected by the choice of marginals.
- In the random vectors examples (where the GR test is not applicable) we saw that in the two 'pairwise independence' examples mBLAW did way better than mHHG. Next, in all the copula examples mBLAW performed better than the two other tests. In particular, it performed drastically better in the case of Cauchy margins, again illustrating its robustness in the presence of extreme values.

To summarize, we extended the HHG test for multivariate independence testing because few tests are consistent to detect mutual dependence between more than two vectors. Then, through power simulations we realized that, especially for the version of the test using ranks, this new test is pretty powerful.

5.3. Limitations

The main limitation of the proposed test is the computation time which is high, especially for large samples sizes N. To be more precise, in the multivariate case if we make the assumption that d (the number of vectors) is a lot smaller then N (which is a reasonable assumption in most applications), the computation of the test statistic T is done in approximately $d \times N^3$ operations. Furthermore, when using a permutation method (with number of permutations n_p), the p-value is calculated by basically redoing the same thing n_p times, hence needing a number of operations of order $n_p \times d \times N^3$. However, this issue could be partially solved using the arguments presented in the next section 5.4.

Finally, as in any statistical test, the outcome we obtain is binary: we reject H_0 or we do not. This gives a very “black and white” picture of the situation. In the case H_0 is rejected the procedure gives no additional information about the dependence structure of the data. Of course, modeling is not the scope of this research. However it is worth mentioning that, upon discovering there is dependence between some variables, one might want to understand more precisely *how* these variables are related, possibly with the objective of *explaining* some of them using the others.

5.4. Possible follow-up work

For reasons explained in section 3.1 and in sight of the power simulations conducted, using ranked-data seems like a good idea. Here we present a further argument that supports using ranks, at least in the case of $X^{(1)}, \cdots, X^{(d)}$ continuous random variables. Although we did not fully develop this idea in the present research, it could be brought forward in future work.

It is well known that for X a continuous random variable and $F(\cdot)$ its cumulative distribution function, we have:

$$F(X) \sim U[0,1]$$

Therefore, if we apply the test to the empirical distribution functions of our observations, $F_N[X^{(1)}], F_N[X^{(2)}], \cdots, F_N[X^{(d)}]$ (which are functions of the ranks) rather then on the observations themselves, we now know the distribution of the marginals: they are asymptotically Uniform $(0,1)$. This has the consequence that the expected counts $E(t)$ in a particular cell of our contingency tables $t = (t_1, t_2, \cdots, t_d)$ is now known theoretically, and does not need to be estimated. In other words, in section 4.2 we used to work with the empirical expected counts $\hat{E}(t)$:

$$\hat{E}(t_1, \cdots, t_d) = \frac{1}{(N-2)^{d-1}} \times \prod_{v=1}^{d} \left(\sum_{\substack{k=1 \\ k \notin \{i,j\}}}^{N} \mathbb{1}(I_k^{(v)} = t_v) \right).$$

We could now use theoretical expected counts:

$$E(t_1, \cdots, t_d) = N \times \mathrm{p}(t_1, \cdots, t_d)$$

where $\mathrm{p}(t_1, \cdots, t_d)$ is the theoretical probability for a point to fall in category t, under H_0.

Now, recall our partial test statistic $S(\cdot, \cdot)$ calculated for two specific points i, j, denoted $S(i,j)$:

$$S(i,j) = \sum_t \frac{[A(t) - E(t)]^2}{E(t)}.$$

Unless the two chosen points i and j are exactly equal in one or more of their d components (which should in principle never happen for continuous random variables), $E(t)$ will never be 0 and this statistic will always be computable. This is not the case if we use the empirical $\hat{E}(t)$. Furthermore, this is now asymptotically a χ^2 statistic with $2^d - 1$ degrees of freedom, compared to $2^d - d - 1$ when we had to estimate the $E(t)$ empirically.

We believe the addition of these non-zero $S(\cdot,\cdot)$ to the overall statistic $T = \sum_{i \neq j} S(i,j)$, as well as the increased number of degrees of freedom might enhance the power of the test.

Furthermore, and perhaps most importantly, recall that we do not know the distribution of the test statistic T. For small samples sizes N this is not really a concern because the permutation method works well and does not take too long to compute. However for large sample sizes N, their might be issues with computation time. This is because the permutation method we use basically means that we repeat the same thing a large number of times. Of course, if we knew the distribution of $T|H_0$, then we wouldn't need the permutation method and the computation time would decrease drastically. Hence, there is interest in finding parametrical approximations to the distribution of $T|H_0$, or at the least in finding a good approximation of its high quantiles. But in the original version of the test, $T|H_0$ actually depends of the marginals of $X = X^{(1)}$ and $Y = X^{(2)}$. Hence this task might not be feasible. If on the contrary we use ranks, $T|H_0$ always has the same (asymptotical) distribution. Then there is hope in finding a good parametrical fit to $T|H_0$. Alternatively, with simulations we could built an (empirical) table of the quantiles of $T|H_0$ for a series of N, and extrapolate the quantiles for in-between values of N. Again, using this table instead of the permutation method would greatly reduce the computation time of the test.

Bibliography

[1] Anscombe, F. J. (1973). Graphs in statistical analysis. *The American Statistician*, 27(1):17–21.

[2] Bakirov, N. K., Rizzo, M. L., and Székely, G. J. (2006). A multivariate nonparametric test of independence. *Journal of Multivariate Analysis*, 97(8):1742–1756.

[3] Beran, R., Bilodeau, M., and Lafaye de Micheaux, P. (2007). Nonparametric tests of independence between random vectors. *Journal of Multivariate Analysis*, 98(9):1805–1824.

[4] Bilodeau, M. and Lafaye de Micheaux, P. (2005). A multivariate empirical characteristic function test of independence with normal marginals. *Journal of Multivariate Analysis*, 95(2):345–369.

[5] Blomqvist, N. (1950). On a measure of dependence between two random variables. *The Annals of Mathematical Statistics*, pages 593–600.

[6] Deheuvels, P. (1981). An asymptotic decomposition for multivariate distribution-free tests of independence. *Journal of Multivariate Analysis*, 11(1):102–113.

[7] Duan, J. (2015). *An introduction to stochastic dynamics*, volume 51. Cambridge University Press.

[8] Eddelbuettel, D., François, R., Allaire, J., Chambers, J., Bates, D., and Ushey, K. (2011). Rcpp: Seamless r and c++ integration. *Journal of Statistical Software*, 40(8):1–18.

[9] Galton, F. (1888). Co-relations and their measurement, chiefly from anthropometric data. *Proceedings of the Royal Society of London*, 45(273-279):135–145.

[10] Genest, C. and Rémillard, B. (2004). Test of independence and randomness based on the empirical copula process. *Test*, 13(2):335–369.

[11] Gieser, P. W. and Randles, R. H. (1997). A nonparametric test of independence between two vectors. *Journal of the American Statistical Association*, 92(438):561–567.

[12] Gorfine, M., Heller, R., and Heller, Y. (2012). Comment on detecting novel associations in large data sets. *Unpublished manuscript.*

[13] Gretton, A., Bousquet, O., Smola, A., and Schölkopf, B. (2005). Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer.

[14] Gretton, A., Fukumizu, K., Teo, C. H., Song, L., Schölkopf, B., and Smola, A. J. (2007). A kernel statistical test of independence. In *Advances in neural information processing systems*, pages 585–592.

[15] Heller, R., Heller, Y., and Gorfine, M. (2012). A consistent multivariate test of association based on ranks of distances. *Biometrika*, 100(2):503–510.

[16] Hoeffding, W. (1948). A non-parametric test of independence. *The Annals of Mathematical Statistics*, pages 546–557.

[17] Kallenberg, W. C. and Ledwina, T. (1999). Data-driven rank tests for independence. *Journal of the American Statistical Association*, 94(445):285–301.

[18] Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

[19] Kojadinovic, I. and Holmes, M. (2009). Tests of independence among continuous random vectors based on cramér–von mises functionals of the empirical copula process. *Journal of Multivariate Analysis*, 100(6):1137–1154.

[20] Lafaye De Micheaux, P., Drouilhet, R., and Liquet, B. (2011). *Le logiciel R: Maitriser le langage-Effectuer des analyses statistiques.* Springer Science & Business Media.

[21] Lee Rodgers, J. and Nicewander, W. A. (1988). Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66.

[22] Pearson, K. (1896). Mathematical contributions to the theory of evolution.–on a form of spurious correlation which may arise when indices are used in the measurement of organs. *Proceedings of the royal society of london*, 60(359-367):489–498.

[23] Pfister, N., Bühlmann, P., Schölkopf, B., and Peters, J. (2016). Kernel-based tests for joint independence. *arXiv preprint arXiv:1603.00285.*

[24] Puri, M. and Sen, P. (1971). Nonparametric methods in multivariate analysis.

[25] Rényi, A. (1959). On measures of dependence. *Acta mathematica hungarica*, 10(3-4):441–451.

[26] Reshef, D. N., Reshef, Y. A., Finucane, H. K., Grossman, S. R., McVean, G., Turnbaugh, P. J., Lander, E. S., Mitzenmacher, M., and Sabeti, P. C. (2011). Detecting novel associations in large data sets. *science*, 334(6062):1518–1524.

[27] Resnick, S. I. (2013). *A probability path.* Springer Science & Business Media.

[28] Romano, J. P. and Siegel, A. F. (1986). *Counterexamples in probability and statistics*. CRC Press.

[29] Rosenthal, J. S. (2006). *A first look at rigorous probability theory*. World Scientific.

[30] Sen, A. and Srivastava, M. (2012). *Regression analysis: theory, methods, and applications*. Springer Science & Business Media.

[31] Serfling, R. J. (2009). *Approximation theorems of mathematical statistics*, volume 162. John Wiley & Sons.

[32] Simon, N. and Tibshirani, R. (2014). Comment on" detecting novel associations in large data sets" by reshef et al, science dec 16, 2011. *arXiv preprint arXiv:1401.7645*.

[33] Spearman, C. (1904). The proof and measurement of association between two things. *The American journal of psychology*, 15(1):72–101.

[34] Székely, G. J., Rizzo, M. L., Bakirov, N. K., et al. (2007). Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794.

[35] Székely, G. J., Rizzo, M. L., et al. (2009). Brownian distance covariance. *The annals of applied statistics*, 3(4):1236–1265.

[36] Taskinen, S., Oja, H., and Randles, R. H. (2005). Multivariate nonparametric tests of independence. *Journal of the American Statistical Association*, 100(471):916–925.

[37] Um, Y. and Randles, R. H. (2001). A multivariate nonparametric test of independence among many vectors. *Journal of Nonparametric Statistics*, 13(5):699–708.

[38] Wald, A. and Brookner, R. (1941). On the distribution of wilks' statistic for testing the independence of several groups of variates. *The Annals of Mathematical Statistics*, 12(2):137–152.

[39] Wilding, G. E. and Mudholkar, G. S. (2008). Empirical approximations for hoeffding's test of bivariate independence using two weibull extensions. *Statistical Methodology*, 5(2):160–170.

[40] Wilks, S. (1935). On the independence of k sets of normally distributed statistical variables. *Econometrica, Journal of the Econometric Society*, pages 309–326.

Appendix A

PROOFS OF SECONDARY RESULTS

A.1. Definition of independence

Theorem 4.2.1 in Resnick [27] states a 'well known' fact: random variables are independent if and only if their joint distribution function is the product of the marginal distribution functions. We want to prove a slightly different result, and to do so we get inspiration from the proof of theorem 4.2.1 in Resnick [27].

Theorem A.1.1. *A finite collection of random variables $X_1, \cdots, X_k$ is independent iff*

$$P[a_1 \leq X_1 \leq b_1, \cdots, a_k \leq X_k \leq b_k] = \prod_{i=1}^{k} P[a_i \leq X_i \leq b_i], \quad \forall a_i, b_i \in \mathbb{R} \quad (A.1.1)$$

Proof.
($\Rightarrow$) This part is easy. If $X_1, \cdots, X_k$ are independent it means their induced sigma-fields $\sigma(X_1), \cdots, \sigma(X_k)$ are independent. Since

$$[a_i \leq X_i \leq b_i] \in \sigma(X_i),$$

all the events $[a_i \leq X_i \leq b_i], i = 1, \cdots, k$ are independent, and the result follows by definition of independent events.
($\Leftarrow$) We define the following classes of subsets:

$$\mathcal{C} = \{[a_t \leq b_t], a_t, b_t \in \mathbb{R}\},$$

which is the class of closed intervals on $\mathbb{R}$.

$$\mathcal{C}_t = \{[a_t \leq X_t \leq b_t], a_t, b_t \in \mathbb{R}\},$$

which is the class of events 'random variable X_t is within a closed interval on $\mathbb{R}$'.

Then

(i) $\mathcal{C}_t$ is a π-system (class closed under finite intersection) since

$$[a_i \leq X_t \leq b_i] \bigcap [a_j \leq X_t \leq b_j] = [a_i \wedge a_j \leq X_t \leq b_i \wedge b_j] \in \mathcal{C}_t.$$

(ii) $\sigma(\mathcal{C}_t) = \sigma(X_t)$, since $\sigma(\mathcal{C})$ generates the Borel subsets of $\mathbb{R}$:

$$\begin{aligned}\sigma(\mathcal{C}_t) = \sigma([X_t \in B], B \in \mathcal{C}) &= \sigma([X_t^{-1}(B)], B \in \mathcal{C}) \\ &= \sigma((X_t^{-1}(\mathcal{C})) = X_t^{-1}(\sigma(\mathcal{C})) \\ &= X_t^{-1}(\mathcal{B}(\mathbb{R})) = \sigma(X_t)\end{aligned}$$

Now, A.1.1 means that the classes $\{\mathcal{C}_t\}$ are independent. Then, because these classes are π-systems, we can use Theorem 4.1.1 in Resnick [27] to establish that $\{\sigma(\mathcal{C}_t) = \sigma(X_t)\}$ are independent.

A.2. Proof that $X \perp Y \iff F_X(X) \perp F_Y(Y)$

We need the following theorem taken from Rosenthal [29], p. 32:

Theorem A.2.1. *Let X and Y be independent random variables. Let $f, g : \mathbb{R} \to \mathbb{R}$ be Borel-measurable functions. Then the random variables $f(X)$ and $g(X)$ are independent.*

Also, one way to state a function is Borel-measurable is the following.

Definition A.2.1. *A function $\phi : \mathbb{R} \to \mathbb{R}$ is Borel measurable if*

$$\forall y \in \mathbb{R} \quad \{x \in \mathbb{R} : \phi(x) \leq y\} \in \mathcal{B}.$$

With this in mind, we prove each side of the 'if and only if' statement that interests us.

($\Rightarrow$) Any cumulative distribution function $F(\cdot)$ is monotone increasing. Then, $\forall y \in \mathbb{R}$ the set

$$\{x \in \mathbb{R} : F(x) \leq y\}$$

Is of the form $(-\infty, \lambda] \in \mathcal{B}$ or of the form $(-\infty, \lambda) \in \mathcal{B}$ for some $\lambda \in \mathbb{R}$. In other words, $F(\cdot)$ is Borel-measurable, so the result is immediate from theorem A.2.1.

($\Leftarrow$) Any quantile function $F^{-1}(\cdot)$, or if you prefer the generalized inverse of $F(\cdot)$, is monotone increasing. Hence it is Borel measurable for the same reason that $F(\cdot)$ is. But since we are restricted to the case of continuous random variables, $F^{-1}(F(t)) = t$. This completes the proof.

A.3. Proof that $\sum_t e(t) = M^d$

In section 4.2 we used $\sum_t e(t) = M^d$. Here is the justification. Recall that for a sample of size N, $M = N - 2$. Recall also that $t = (t_1, \ldots, t_d)$. We had:

$$e(t) = \prod_{v=1}^{d} \left(\sum_{\substack{k=1 \\ k \notin \{i,j\}}}^{N} \mathbb{1}(I_k^{(v)} = t_v) \right).$$

For simplification, let us denote:

$$\sum_{\substack{k=1 \\ k \notin \{i,j\}}}^{N} \mathbb{1}(I_k^{(v)} = t_v) \equiv A^v(t_v)$$

so that:

$$e(t) = A^1(t_1) \times A^2(t_2) \times \ldots \times A^d(t_d)$$

or in a slight abuse of notation:

$$e(t) = A^1(t) \times A^2(t) \times \ldots \times A^d(t).$$

Recall that each t_v can only take two values, $t_v \in \{0, 1\}$. Because $A^v(t_v)$ is the number of points out of M to fall in category t_v, with only two possibilities in total, we always have:

$$A^v(0) + A^v(1) = M.$$

Then:

$$\begin{aligned} \sum_t e(t) =& A^1(0) \cdot \left[\sum_t A^2(t) \times \ldots \times A^d(t) \right] + A^1(1) \cdot \left[\sum_t A^2(t) \times \ldots \times A^d(t) \right] \\ =& \left[A^1(0) + (M - A^1(0)) \right] \cdot \left[\sum_t A^2(t) \times \ldots \times A^d(t) \right] \\ =& M \cdot \sum_t A^2(t) \times \ldots \times A^d(t). \end{aligned}$$

And repeating the argument another $d - 1$ times we get:

$$e(t) = M^d.$$

Appendix B

TABLES OF EMPIRICAL POWERS

In this section we display the empirical power for all examples tested.

B.1. Bivariate version of the test

B.1.1. Random variables examples

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	5.1	0.2	5.1	0.2	5.0	0.2	5.1	0.2
20	5.0	0.2	5.0	0.2	4.8	0.2	5.0	0.2
30	5.4	0.2	5.2	0.2	5.3	0.2	5.3	0.2
40	5.5	0.2	5.5	0.2	5.2	0.2	5.1	0.2
50	5.0	0.2	5.0	0.2	4.7	0.2	4.8	0.2
60	5.0	0.2	5.1	0.2	5.1	0.2	5.1	0.2
70	5.1	0.2	5.3	0.2	4.8	0.2	5.0	0.2
80	4.8	0.2	4.7	0.2	4.7	0.2	4.7	0.2
90	5.2	0.2	5.1	0.2	5.0	0.2	5.1	0.2
100	4.8	0.2	4.5	0.2	4.8	0.2	4.7	0.2

Tab. B.1. Power in the case of four independent clouds

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	17.3	0.4	21.4	0.4	10.1	0.3	28.0	0.4
20	59.4	0.5	65.9	0.5	27.8	0.4	64.8	0.5
30	89.6	0.3	90.8	0.3	56.9	0.5	85.6	0.4
40	98.4	0.1	98.3	0.1	78.9	0.4	95.0	0.2
50	99.9	0.0	99.9	0.0	93.5	0.2	98.8	0.1
60	100.0	0.0	100.0	0.0	98.0	0.1	99.6	0.1
70	100.0	0.0	100.0	0.0	99.6	0.1	100.0	0.0
80	100.0	0.0	100.0	0.0	99.9	0.0	100.0	0.0
90	100.0	0.0	100.0	0.0	100.0	0.0	100.0	0.0
100	100.0	0.0	100.0	0.0	100.0	0.0	100.0	0.0

Tab. B.2. Power in the case the W-shape dependence

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	3.4	0.2	4.9	0.2	1.9	0.1	6.5	0.2
20	6.7	0.3	13.4	0.3	2.6	0.2	14.5	0.4
30	14.3	0.4	28.3	0.5	3.0	0.2	26.2	0.4
40	26.1	0.4	48.6	0.5	3.8	0.2	40.1	0.5
50	40.4	0.5	66.4	0.5	4.8	0.2	54.5	0.5
60	54.4	0.5	80.2	0.4	6.0	0.2	66.8	0.5
70	68.1	0.5	89.3	0.3	7.4	0.3	77.5	0.4
80	79.2	0.4	94.8	0.2	9.6	0.3	85.3	0.4
90	87.9	0.3	97.6	0.2	12.2	0.3	90.9	0.3
100	93.2	0.3	99.1	0.1	16.4	0.4	94.4	0.2

Tab. B.3. Power in the case of the Diamond dependence

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	17.8	0.4	19.1	0.4	15.0	0.4	20.8	0.4
20	53.8	0.5	50.8	0.5	31.2	0.5	47.6	0.5
30	84.5	0.4	79.1	0.4	52.2	0.5	73.3	0.4
40	96.8	0.2	92.8	0.3	71.2	0.5	88.2	0.3
50	99.5	0.1	98.3	0.1	85.4	0.4	95.7	0.2
60	99.9	0.0	99.6	0.1	93.4	0.2	98.6	0.1
70	100.0	0.0	99.9	0.0	97.9	0.1	99.6	0.1
80	100.0	0.0	100.0	0.0	99.4	0.1	100.0	0.0
90	100.0	0.0	100.0	0.0	99.8	0.0	100.0	0.0
100	100.0	0.0	100.0	0.0	99.9	0.0	100.0	0.0

Tab. B.4. Power in the case of the Parabola dependence

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	28.6	0.5	31.4	0.5	11.7	0.3	20.0	0.4
20	78.3	0.4	74.9	0.4	15.7	0.4	53.9	0.5
30	98.8	0.1	96.2	0.2	20.4	0.4	84.9	0.4
40	100.0	0.0	99.7	0.1	26.9	0.4	97.1	0.2
50	100.0	0.0	100.0	0.0	35.4	0.5	99.7	0.1
60	100.0	0.0	100.0	0.0	46.6	0.5	100.0	0.0
70	100.0	0.0	100.0	0.0	58.0	0.5	100.0	0.0
80	100.0	0.0	100.0	0.0	70.0	0.5	100.0	0.0
90	100.0	0.0	100.0	0.0	81.5	0.4	100.0	0.0
100	100.0	0.0	100.0	0.0	90.1	0.3	100.0	0.0

Tab. B.5. Power in the case of the Two Parabolas dependence

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	6.9	0.3	8.8	0.3	2.8	0.2	10.6	0.3
20	34.7	0.5	30.0	0.5	3.4	0.2	25.2	0.4
30	79.8	0.4	64.6	0.5	5.1	0.2	48.8	0.5
40	97.7	0.1	89.8	0.3	6.6	0.2	76.1	0.4
50	99.9	0.0	98.0	0.1	7.5	0.3	91.4	0.3
60	100.0	0.0	99.8	0.0	10.7	0.3	97.5	0.2
70	100.0	0.0	100.0	0.0	13.9	0.3	99.6	0.1
80	100.0	0.0	100.0	0.0	19.6	0.4	99.9	0.0
90	100.0	0.0	100.0	0.0	24.8	0.4	100.0	0.0
100	100.0	0.0	100.0	0.0	32.9	0.5	100.0	0.0

Tab. B.6. Power in the case of the Circle dependence

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	17.8	0.4	17.2	0.4	28.8	0.5	21.1	0.4
20	43.1	0.5	37.7	0.5	55.4	0.5	40.6	0.5
30	64.9	0.5	56.6	0.5	74.0	0.4	58.2	0.5
40	79.5	0.4	72.6	0.4	86.3	0.3	72.2	0.4
50	89.4	0.3	83.3	0.4	93.7	0.2	82.7	0.4
60	94.5	0.2	90.2	0.3	96.9	0.2	89.3	0.3
70	97.5	0.2	95.1	0.2	98.8	0.1	94.4	0.2
80	98.8	0.1	97.1	0.2	99.4	0.1	96.2	0.2
90	99.2	0.1	98.4	0.1	99.7	0.1	98.0	0.1
100	99.8	0.0	99.2	0.1	99.9	0.0	99.0	0.1

Tab. B.7. Power in the case of the Linear dependence

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	9.3	0.3	9.0	0.3	15.0	0.4	10.0	0.3
20	16.6	0.4	16.3	0.4	25.7	0.4	15.5	0.4
30	28.0	0.4	26.1	0.4	37.5	0.5	21.9	0.4
40	37.9	0.5	35.3	0.5	47.6	0.5	28.3	0.5
50	49.3	0.5	45.5	0.5	57.9	0.5	36.1	0.5
60	59.9	0.5	55.8	0.5	66.4	0.5	42.0	0.5
70	69.1	0.5	64.9	0.5	75.3	0.4	49.9	0.5
80	76.6	0.4	72.3	0.4	80.5	0.4	58.1	0.5
90	82.4	0.4	78.4	0.4	85.2	0.4	63.5	0.5
100	87.9	0.3	84.2	0.4	88.9	0.3	69.5	0.5

TAB. B.8. Power in the case of the Exponential dependence

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	18.2	0.4	17.6	0.4	28.2	0.5	26.7	0.4
20	45.0	0.5	39.3	0.5	55.9	0.5	51.9	0.5
30	71.6	0.5	64.1	0.5	77.5	0.4	73.3	0.4
40	87.3	0.3	80.3	0.4	89.7	0.3	85.9	0.3
50	94.9	0.2	91.0	0.3	95.8	0.2	93.9	0.2
60	98.3	0.1	96.0	0.2	98.4	0.1	97.4	0.2
70	99.2	0.1	98.1	0.1	99.4	0.1	98.8	0.1
80	99.8	0.0	99.3	0.1	99.8	0.0	99.5	0.1
90	100.0	0.0	99.8	0.0	100.0	0.0	99.8	0.0
100	100.0	0.0	99.9	0.0	100.0	0.0	99.9	0.0

TAB. B.9. Power in the case of the Sine dependence

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	10.9	0.3	10.6	0.3	18.1	0.4	12.1	0.3
20	23.7	0.4	19.8	0.4	32.4	0.5	19.8	0.4
30	35.8	0.5	30.4	0.5	45.2	0.5	28.5	0.5
40	48.7	0.5	40.8	0.5	55.6	0.5	36.0	0.5
50	59.0	0.5	49.6	0.5	66.5	0.5	43.7	0.5
60	69.4	0.5	59.0	0.5	74.7	0.4	51.4	0.5
70	75.2	0.4	65.7	0.5	80.3	0.4	57.8	0.5
80	81.3	0.4	72.1	0.4	85.5	0.4	63.4	0.5
90	87.0	0.3	79.0	0.4	90.2	0.3	71.4	0.5
100	90.1	0.3	83.0	0.4	92.8	0.3	75.4	0.4

TAB. B.10. Power in the case of the Clayton (0.6) copula with Normal margins

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	11.4	0.3	10.2	0.3	10.1	0.3	11.5	0.3
20	23.2	0.4	17.3	0.4	16.5	0.4	19.1	0.4
30	36.4	0.5	25.2	0.4	24.5	0.4	27.3	0.4
40	47.6	0.5	33.3	0.5	31.5	0.5	34.1	0.5
50	59.5	0.5	42.2	0.5	39.2	0.5	41.5	0.5
60	67.9	0.5	50.7	0.5	47.6	0.5	50.0	0.5
70	76.6	0.4	59.1	0.5	55.2	0.5	57.3	0.5
80	82.0	0.4	65.9	0.5	61.9	0.5	62.5	0.5
90	86.3	0.3	72.0	0.4	67.8	0.5	69.4	0.5
100	90.5	0.3	76.9	0.4	72.8	0.4	74.1	0.4

Tab. B.11. Power in the case of the Clayton (0.6) copula with Exponential margins

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	11.7	0.3	9.9	0.3	9.9	0.3	11.1	0.3
20	23.0	0.4	13.8	0.3	13.4	0.3	14.0	0.3
30	35.5	0.5	19.9	0.4	15.7	0.4	17.6	0.4
40	48.6	0.5	26.5	0.4	18.2	0.4	21.4	0.4
50	59.4	0.5	32.0	0.5	20.7	0.4	26.0	0.4
60	68.0	0.5	35.9	0.5	20.8	0.4	29.8	0.5
70	75.8	0.4	41.0	0.5	22.8	0.4	33.2	0.5
80	81.5	0.4	47.3	0.5	23.9	0.4	38.3	0.5
90	86.1	0.3	52.4	0.5	26.0	0.4	42.7	0.5
100	90.7	0.3	56.6	0.5	27.5	0.4	46.4	0.5

Tab. B.12. Power in the case of the Clayton (0.6) copula with Cauchy margins

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	14.8	0.4	14.1	0.3	24.0	0.4	15.9	0.4
20	31.8	0.5	27.3	0.4	40.7	0.5	27.1	0.4
30	49.6	0.5	42.4	0.5	57.7	0.5	38.7	0.5
40	64.8	0.5	56.0	0.5	70.8	0.5	50.1	0.5
50	75.9	0.4	66.8	0.5	80.6	0.4	60.0	0.5
60	84.1	0.4	76.4	0.4	87.7	0.3	69.3	0.5
70	89.0	0.3	82.7	0.4	92.0	0.3	76.4	0.4
80	92.8	0.3	87.5	0.3	94.7	0.2	81.5	0.4
90	95.8	0.2	91.7	0.3	96.9	0.2	86.9	0.3
100	97.5	0.2	94.6	0.2	98.4	0.1	90.3	0.3

Tab. B.13. Power in the case of the Gumbel (1.4) copula with Normal margins

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	14.7	0.4	16.5	0.4	25.7	0.4	18.3	0.4
20	31.9	0.5	34.7	0.5	44.7	0.5	32.6	0.5
30	49.4	0.5	51.5	0.5	59.5	0.5	43.5	0.5
40	64.0	0.5	66.0	0.5	72.9	0.4	54.1	0.5
50	75.5	0.4	76.3	0.4	81.3	0.4	64.7	0.5
60	83.0	0.4	83.1	0.4	87.5	0.3	71.9	0.4
70	89.1	0.3	89.0	0.3	92.2	0.3	78.9	0.4
80	93.2	0.3	93.2	0.3	95.0	0.2	84.1	0.4
90	95.6	0.2	95.5	0.2	96.5	0.2	88.6	0.3
100	97.3	0.2	97.3	0.2	98.0	0.1	91.2	0.3

TAB. B.14. Power in the case of the Gumbel (1.4) copula with Exponential margins

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	14.5	0.4	12.3	0.3	11.3	0.3	13.9	0.3
20	32.5	0.5	21.5	0.4	14.9	0.4	21.0	0.4
30	49.9	0.5	31.0	0.5	17.4	0.4	27.4	0.4
40	64.2	0.5	39.8	0.5	21.0	0.4	34.5	0.5
50	74.8	0.4	49.0	0.5	22.8	0.4	41.1	0.5
60	83.1	0.4	55.9	0.5	24.7	0.4	48.2	0.5
70	89.7	0.3	63.3	0.5	26.5	0.4	54.9	0.5
80	93.2	0.3	68.9	0.5	27.4	0.4	60.6	0.5
90	96.0	0.2	75.8	0.4	30.4	0.5	66.8	0.5
100	97.2	0.2	79.9	0.4	33.5	0.5	70.2	0.5

TAB. B.15. Power in the case of the Gumbel (1.4) copula with Cauchy margins

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	11.6	0.3	10.4	0.3	20.1	0.4	13.0	0.3
20	24.6	0.4	19.5	0.4	35.8	0.5	20.7	0.4
30	37.4	0.5	28.8	0.5	50.0	0.5	28.8	0.5
40	51.1	0.5	39.6	0.5	62.4	0.5	37.7	0.5
50	62.0	0.5	48.6	0.5	72.8	0.4	45.2	0.5
60	71.1	0.5	57.3	0.5	81.1	0.4	54.1	0.5
70	78.8	0.4	65.2	0.5	86.8	0.3	61.8	0.5
80	85.1	0.4	71.7	0.5	91.3	0.3	67.3	0.5
90	89.5	0.3	78.3	0.4	94.4	0.2	73.6	0.4
100	92.5	0.3	83.0	0.4	96.2	0.2	77.5	0.4

TAB. B.16. Power in the case of the Normal copula ($\rho = 0.4$) with Normal margins

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	11.2	0.3	11.8	0.3	18.8	0.4	14.6	0.4
20	24.5	0.4	22.7	0.4	32.6	0.5	24.5	0.4
30	38.4	0.5	33.9	0.5	45.2	0.5	33.7	0.5
40	51.7	0.5	46.3	0.5	56.8	0.5	44.4	0.5
50	62.0	0.5	55.0	0.5	67.0	0.5	51.4	0.5
60	70.4	0.5	63.1	0.5	74.3	0.4	57.6	0.5
70	78.9	0.4	71.5	0.5	82.2	0.4	66.4	0.5
80	84.8	0.4	79.0	0.4	87.4	0.3	72.7	0.4
90	88.6	0.3	83.2	0.4	90.3	0.3	76.6	0.4
100	92.7	0.3	88.2	0.3	93.4	0.2	83.0	0.4

TAB. B.17. Power in the case of the Normal copula ($\rho = 0.4$) with Exponential margins

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	11.7	0.3	8.8	0.3	9.0	0.3	9.9	0.3
20	24.6	0.4	12.9	0.3	11.2	0.3	13.7	0.3
30	38.2	0.5	16.8	0.4	12.8	0.3	17.1	0.4
40	50.2	0.5	21.2	0.4	13.5	0.3	19.9	0.4
50	62.5	0.5	27.2	0.4	14.8	0.4	25.0	0.4
60	70.9	0.5	30.1	0.5	15.0	0.4	28.0	0.4
70	79.1	0.4	35.9	0.5	16.6	0.4	32.9	0.5
80	84.4	0.4	40.0	0.5	17.1	0.4	36.3	0.5
90	89.4	0.3	44.6	0.5	17.2	0.4	41.8	0.5
100	92.3	0.3	50.0	0.5	18.5	0.4	44.5	0.5

TAB. B.18. Power in the case of the Normal copula ($\rho = 0.4$) with Cauchy margins

B.1.2. Random vectors examples

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	17.9	0.4	11.4	0.3	5.9	0.2	13.3	0.3
20	52.7	0.5	32.3	0.5	9.9	0.3	33.9	0.5
30	85.5	0.4	59.4	0.5	15.4	0.4	60.6	0.5
40	97.4	0.2	81.2	0.4	25.8	0.4	83.4	0.4
50	99.8	0.0	93.6	0.2	38.6	0.5	95.8	0.2
60	100.0	0.0	98.9	0.1	53.0	0.5	99.2	0.1
70	100.0	0.0	99.7	0.1	67.8	0.5	100.0	0.0
80	100.0	0.0	100.0	0.0	79.3	0.4	100.0	0.0
90	100.0	0.0	100.0	0.0	88.9	0.3	100.0	0.0
100	100.0	0.0	100.0	0.0	96.0	0.2	100.0	0.0

TAB. B.19. Power in the case of vectors for the ‘log’ dependence

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	8.7	0.3	21.6	0.4	13.9	0.3	22.0	0.4
20	18.9	0.4	55.7	0.5	18.8	0.4	36.2	0.5
30	31.4	0.5	78.4	0.4	22.6	0.4	48.4	0.5
40	45.6	0.5	90.5	0.3	26.3	0.4	58.4	0.5
50	60.1	0.5	96.9	0.2	29.8	0.5	68.9	0.5
60	71.6	0.5	98.7	0.1	32.6	0.5	77.4	0.4
70	81.0	0.4	99.6	0.1	35.7	0.5	84.6	0.4
80	88.3	0.3	99.9	0.0	38.2	0.5	89.6	0.3
90	93.1	0.3	100.0	0.0	42.8	0.5	93.6	0.2
100	95.5	0.2	100.0	0.0	46.3	0.5	95.8	0.2

TAB. B.20. Power in the case of vectors for the ‘epsilon’ dependence

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	6.6	0.2	17.4	0.4	15.0	0.4	20.5	0.4
20	10.0	0.3	44.8	0.5	22.9	0.4	37.6	0.5
30	15.8	0.4	68.7	0.5	30.2	0.5	53.0	0.5
40	23.8	0.4	85.2	0.4	39.9	0.5	67.7	0.5
50	31.6	0.5	93.4	0.2	48.5	0.5	80.4	0.4
60	42.2	0.5	97.3	0.2	56.9	0.5	88.7	0.3
70	51.4	0.5	98.7	0.1	65.4	0.5	93.8	0.2
80	62.2	0.5	99.6	0.1	72.5	0.4	97.1	0.2
90	68.5	0.5	99.9	0.0	78.3	0.4	98.8	0.1
100	77.5	0.4	99.9	0.0	84.6	0.4	99.3	0.1

TAB. B.21. Power in the case of vectors for the ‘quadratic’ dependence

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	6.1	0.2	6.7	0.3	7.3	0.3	11.2	0.3
20	12.8	0.3	11.2	0.3	11.7	0.3	22.0	0.4
30	24.7	0.4	16.1	0.4	17.5	0.4	40.3	0.5
40	42.0	0.5	21.4	0.4	25.3	0.4	60.0	0.5
50	64.1	0.5	27.0	0.4	35.9	0.5	79.8	0.4
60	85.3	0.4	36.2	0.5	49.4	0.5	92.5	0.3
70	95.3	0.2	44.4	0.5	63.0	0.5	97.7	0.2
80	99.2	0.1	54.4	0.5	75.2	0.4	99.3	0.1
90	99.9	0.0	63.7	0.5	86.3	0.3	99.9	0.0
100	100.0	0.0	72.7	0.4	92.9	0.3	100.0	0.0

Tab. B.22. Power in the case of the pairwise independence example

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	7.4	0.3	5.5	0.2	5.6	0.2	5.3	0.2
20	12.2	0.3	5.3	0.2	6.1	0.2	5.4	0.2
40	24.2	0.4	6.2	0.2	6.8	0.3	6.0	0.2
60	39.0	0.5	6.2	0.2	6.1	0.2	6.2	0.2
80	56.2	0.5	7.0	0.3	6.8	0.3	6.8	0.3
100	69.4	0.5	7.7	0.3	7.4	0.3	7.4	0.3
120	80.2	0.4	7.8	0.3	7.3	0.3	7.1	0.3
150	90.1	0.3	8.7	0.3	8.6	0.3	7.9	0.3
200	97.5	0.2	10.4	0.3	9.1	0.3	9.1	0.3
250	99.7	0.1	11.9	0.3	11.2	0.3	9.4	0.3

Tab. B.23. Power in the case of the ‘big noise’ dependence

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	13.5	0.3	13.2	0.3	20.4	0.4	16.4	0.4
20	34.4	0.5	33.0	0.5	39.6	0.5	32.4	0.5
30	54.8	0.5	49.7	0.5	55.7	0.5	46.0	0.5
40	71.0	0.5	63.9	0.5	70.2	0.5	58.2	0.5
50	81.8	0.4	76.5	0.4	80.5	0.4	70.4	0.5
60	89.0	0.3	84.5	0.4	87.6	0.3	78.6	0.4
70	93.7	0.2	90.4	0.3	92.8	0.3	85.3	0.4
80	97.0	0.2	94.2	0.2	96.4	0.2	90.6	0.3
90	98.3	0.1	96.7	0.2	97.6	0.2	93.2	0.3
100	99.0	0.1	98.0	0.1	98.5	0.1	96.1	0.2

Tab. B.24. Power in the case of vectors for the Clayton (0.5) copula with Normal margins

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	13.5	0.3	9.1	0.3	9.8	0.3	11.9	0.3
20	34.4	0.5	17.1	0.4	17.2	0.4	23.6	0.4
30	54.8	0.5	25.3	0.4	26.9	0.4	33.5	0.5
40	71.0	0.5	34.1	0.5	36.9	0.5	43.4	0.5
50	81.8	0.4	43.5	0.5	47.0	0.5	52.8	0.5
60	89.0	0.3	52.5	0.5	57.2	0.5	61.3	0.5
70	93.7	0.2	60.4	0.5	65.0	0.5	68.8	0.5
80	97.0	0.2	67.4	0.5	74.2	0.4	76.8	0.4
90	98.3	0.1	73.6	0.4	79.2	0.4	80.6	0.4
100	99.0	0.1	79.6	0.4	84.8	0.4	85.4	0.4

TAB. B.25. Power in the case of vectors for the Clayton (0.5) copula with Exponential margins

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	13.5	0.3	10.5	0.3	10.4	0.3	12.5	0.3
20	34.4	0.5	18.4	0.4	13.7	0.3	19.9	0.4
30	54.8	0.5	25.8	0.4	15.1	0.4	26.2	0.4
40	71.0	0.5	33.3	0.5	17.5	0.4	33.4	0.5
50	81.8	0.4	40.4	0.5	19.5	0.4	39.3	0.5
60	89.0	0.3	47.8	0.5	22.1	0.4	47.5	0.5
70	93.7	0.2	53.0	0.5	23.0	0.4	52.9	0.5
80	97.0	0.2	58.6	0.5	23.6	0.4	59.0	0.5
90	98.3	0.1	64.0	0.5	25.2	0.4	64.0	0.5
100	99.0	0.1	69.5	0.5	26.4	0.4	69.6	0.5

TAB. B.26. Power in the case of vectors for the Clayton (0.5) copula with Cauchy margins

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	17.5	0.4	17.4	0.4	24.8	0.4	21.4	0.4
20	43.4	0.5	38.0	0.5	45.1	0.5	40.1	0.5
30	64.6	0.5	56.1	0.5	63.3	0.5	55.4	0.5
40	79.8	0.4	71.8	0.4	77.3	0.4	69.3	0.5
50	89.2	0.3	82.2	0.4	86.7	0.3	79.9	0.4
60	94.2	0.2	88.7	0.3	92.4	0.3	87.0	0.3
70	97.3	0.2	93.1	0.3	96.0	0.2	92.3	0.3
80	98.7	0.1	96.5	0.2	98.0	0.1	95.4	0.2
90	99.4	0.1	98.0	0.1	99.0	0.1	97.2	0.2
100	99.8	0.0	98.8	0.1	99.5	0.1	98.5	0.1

TAB. B.27. Power in the case of vectors for the Gumbel (1.3) copula with Normal margins

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	17.5	0.4	18.6	0.4	25.1	0.4	22.9	0.4
20	43.4	0.5	36.5	0.5	42.1	0.5	37.5	0.5
30	64.6	0.5	53.4	0.5	57.8	0.5	50.8	0.5
40	79.8	0.4	67.5	0.5	70.2	0.5	62.0	0.5
50	89.2	0.3	77.2	0.4	79.8	0.4	71.3	0.5
60	94.2	0.2	84.6	0.4	86.8	0.3	79.1	0.4
70	97.3	0.2	89.6	0.3	91.4	0.3	84.8	0.4
80	98.7	0.1	93.4	0.2	95.0	0.2	89.6	0.3
90	99.4	0.1	95.7	0.2	97.0	0.2	93.1	0.3
100	99.8	0.0	97.1	0.2	98.0	0.1	94.8	0.2

TAB. B.28. Power in the case of vectors for the Gumbel (1.3) copula with Exponential margins

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	17.5	0.4	11.9	0.3	10.7	0.3	14.1	0.3
20	43.4	0.5	21.1	0.4	13.3	0.3	22.6	0.4
30	64.6	0.5	29.6	0.5	14.7	0.4	31.1	0.5
40	79.8	0.4	38.0	0.5	17.8	0.4	38.7	0.5
50	89.2	0.3	44.9	0.5	19.3	0.4	45.3	0.5
60	94.2	0.2	52.8	0.5	20.8	0.4	53.4	0.5
70	97.3	0.2	59.0	0.5	22.4	0.4	60.3	0.5
80	98.7	0.1	65.2	0.5	23.9	0.4	67.5	0.5
90	99.4	0.1	71.0	0.5	24.8	0.4	72.4	0.4
100	99.8	0.0	75.2	0.4	25.2	0.4	77.2	0.4

TAB. B.29. Power in the case of vectors for the Gumbel (1.3) copula with Cauchy margins

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	12.1	0.3	11.0	0.3	18.2	0.4	14.1	0.3
20	27.4	0.4	22.7	0.4	34.6	0.5	26.2	0.4
30	46.1	0.5	36.1	0.5	51.2	0.5	38.4	0.5
40	62.4	0.5	49.5	0.5	66.9	0.5	51.5	0.5
50	73.3	0.4	60.2	0.5	77.5	0.4	61.1	0.5
60	82.6	0.4	71.0	0.5	85.5	0.4	70.5	0.5
70	88.0	0.3	77.4	0.4	90.0	0.3	76.9	0.4
80	92.7	0.3	83.8	0.4	94.0	0.2	83.6	0.4
90	95.7	0.2	89.3	0.3	96.5	0.2	88.5	0.3
100	97.2	0.2	92.1	0.3	97.9	0.1	91.2	0.3

TAB. B.30. Power in the case of vectors for the Normal ($\rho = 0.3$) copula with Normal margins

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	12.1	0.3	13.8	0.3	18.3	0.4	17.8	0.4
20	27.4	0.4	26.9	0.4	30.6	0.5	31.1	0.5
30	46.1	0.5	40.3	0.5	44.9	0.5	43.6	0.5
40	62.4	0.5	52.9	0.5	58.1	0.5	54.9	0.5
50	73.3	0.4	64.3	0.5	69.2	0.5	65.1	0.5
60	82.6	0.4	73.0	0.4	78.3	0.4	73.3	0.4
70	88.0	0.3	79.2	0.4	83.8	0.4	78.9	0.4
80	92.7	0.3	85.0	0.4	89.6	0.3	84.4	0.4
90	95.7	0.2	89.8	0.3	93.1	0.3	88.7	0.3
100	97.2	0.2	93.2	0.3	95.4	0.2	91.9	0.3

TAB. B.31. Power in the case of vectors for the Normal ($\rho = 0.3$) copula with Exponential margins

N	BLAW	(sd)	HHG	(sd)	DCOV	(sd)	HSIC	(sd)
10	12.1	0.3	8.2	0.3	7.9	0.3	9.9	0.3
20	27.4	0.4	11.4	0.3	9.7	0.3	13.2	0.3
30	46.1	0.5	15.9	0.4	11.4	0.3	18.4	0.4
40	62.4	0.5	18.9	0.4	11.8	0.3	21.6	0.4
50	73.3	0.4	22.9	0.4	12.4	0.3	25.1	0.4
60	82.6	0.4	26.8	0.4	12.2	0.3	30.5	0.5
70	88.0	0.3	28.4	0.5	13.1	0.3	32.9	0.5
80	92.7	0.3	33.0	0.5	13.8	0.3	37.7	0.5
90	95.7	0.2	36.1	0.5	14.6	0.4	42.1	0.5
100	97.2	0.2	40.8	0.5	14.5	0.4	46.2	0.5

TAB. B.32. Power in the case of vectors for the Normal ($\rho = 0.3$) copula with Cauchy margins

B.2. Multivariate version of the test

B.2.1. Random variables examples

N	mBLAW	(sd)	mHHG	(sd)	BBL	(sd)	mHSIC	(sd)	GR	(sd)
10	7.1	0.3	6.6	0.2	4.9	0.2	16.6	0.4	7.9	0.3
20	13.0	0.3	10.7	0.3	5.1	0.2	34.8	0.5	14.3	0.4
30	23.7	0.4	17.0	0.4	4.6	0.2	60.3	0.5	22.0	0.4
40	39.8	0.5	25.9	0.4	6.0	0.2	80.8	0.4	30.4	0.5
50	58.8	0.5	37.5	0.5	10.0	0.3	93.3	0.2	38.6	0.5
60	77.9	0.4	51.7	0.5	20.5	0.4	98.2	0.1	48.2	0.5
70	90.6	0.3	63.1	0.5	33.0	0.5	99.6	0.1	55.6	0.5
80	97.1	0.2	75.8	0.4	50.0	0.5	100.0	0.0	63.2	0.5
90	99.5	0.1	85.4	0.4	64.4	0.5	100.0	0.0	71.5	0.5
100	99.9	0.0	92.2	0.3	78.6	0.4	100.0	0.0	77.1	0.4

Tab. B.33. Power in the case of the 3D pairwise independent Normals

N	mBLAW	(sd)	mHHG	(sd)	BBL	(sd)	mHSIC	(sd)	GR	(sd)
10	6.9	0.3	7.2	0.3	6.4	0.2	8.5	0.3	7.3	0.3
20	11.3	0.3	11.6	0.3	9.1	0.3	13.8	0.3	8.8	0.3
30	16.9	0.4	16.6	0.4	11.7	0.3	19.1	0.4	9.6	0.3
40	26.1	0.4	25.4	0.4	16.6	0.4	26.8	0.4	11.6	0.3
50	36.8	0.5	34.7	0.5	21.2	0.4	34.0	0.5	13.5	0.3
60	48.0	0.5	44.4	0.5	25.0	0.4	42.4	0.5	14.5	0.4
70	59.1	0.5	55.2	0.5	30.6	0.5	50.9	0.5	17.5	0.4
80	69.2	0.5	64.7	0.5	34.4	0.5	58.6	0.5	20.4	0.4
90	78.0	0.4	73.0	0.4	39.0	0.5	66.6	0.5	22.7	0.4
100	84.8	0.4	80.1	0.4	45.2	0.5	72.8	0.4	25.4	0.4

Tab. B.34. Power in the case of the 3D Cos-Sin dependence

N	mBLAW	(sd)	mHHG	(sd)	BBL	(sd)	mHSIC	(sd)	GR	(sd)
10	12.5	0.3	11.6	0.3	12.1	0.3	11.8	0.3	15.0	0.4
20	27.9	0.4	24.8	0.4	23.4	0.4	22.5	0.4	21.6	0.4
30	47.6	0.5	41.4	0.5	35.0	0.5	34.4	0.5	29.8	0.5
40	65.8	0.5	58.4	0.5	49.1	0.5	47.3	0.5	38.2	0.5
50	80.4	0.4	72.4	0.4	60.8	0.5	61.1	0.5	47.2	0.5
60	89.8	0.3	83.1	0.4	69.9	0.5	71.4	0.5	55.3	0.5
70	95.0	0.2	90.3	0.3	79.1	0.4	80.5	0.4	62.1	0.5
80	97.6	0.2	94.6	0.2	84.8	0.4	87.2	0.3	68.5	0.5
90	99.0	0.1	97.2	0.2	89.4	0.3	92.3	0.3	75.9	0.4
100	99.6	0.1	98.7	0.1	92.4	0.3	95.5	0.2	80.1	0.4

Tab. B.35. Power in the case of the 3D Cos-Exp dependence

N	mBLAW	(sd)	mHHG	(sd)	BBL	(sd)	mHSIC	(sd)	GR	(sd)
10	8.8	0.3	7.9	0.3	8.4	0.3	8.1	0.3	19.1	0.4
20	16.0	0.4	13.0	0.3	15.3	0.4	13.0	0.3	29.3	0.5
30	24.5	0.4	18.6	0.4	21.2	0.4	17.6	0.4	40.1	0.5
40	34.5	0.5	24.8	0.4	29.0	0.5	22.7	0.4	50.8	0.5
50	45.7	0.5	33.0	0.5	37.6	0.5	28.5	0.5	59.9	0.5
60	55.2	0.5	40.2	0.5	43.9	0.5	33.3	0.5	67.5	0.5
70	63.8	0.5	46.9	0.5	50.6	0.5	39.9	0.5	73.8	0.4
80	71.0	0.5	53.2	0.5	57.7	0.5	45.8	0.5	78.7	0.4
90	77.9	0.4	60.6	0.5	63.5	0.5	51.5	0.5	83.9	0.4
100	83.5	0.4	66.5	0.5	70.0	0.5	58.0	0.5	87.7	0.3

Tab. B.36. Power in the case of the 3D linear dependence

N	mBLAW	(sd)	mHHG	(sd)	BBL	(sd)	mHSIC	(sd)	GR	(sd)
10	13.9	0.3	13.0	0.3	12.2	0.3	11.6	0.3	34.2	0.5
20	30.2	0.5	26.4	0.4	21.6	0.4	19.2	0.4	52.1	0.5
30	47.9	0.5	40.7	0.5	30.1	0.5	26.7	0.4	66.7	0.5
40	62.1	0.5	53.3	0.5	41.0	0.5	36.0	0.5	77.7	0.4
50	74.5	0.4	64.8	0.5	50.0	0.5	44.6	0.5	85.5	0.4
60	82.5	0.4	73.9	0.4	59.1	0.5	52.7	0.5	90.1	0.3
70	88.9	0.3	81.4	0.4	66.9	0.5	60.4	0.5	94.3	0.2
80	93.2	0.3	86.5	0.3	74.1	0.4	68.1	0.5	96.7	0.2
90	95.4	0.2	90.6	0.3	78.7	0.4	73.4	0.4	97.7	0.2
100	97.4	0.2	93.9	0.2	83.6	0.4	79.5	0.4	98.6	0.1

Tab. B.37. Power in the case of the 3D Clayton copula (0.5) with Normal margins

N	mBLAW	(sd)	mHHG	(sd)	BBL	(sd)	mHSIC	(sd)	GR	(sd)
10	13.9	0.3	10.8	0.3	12.2	0.3	12.5	0.3	34.2	0.5
20	30.2	0.5	20.8	0.4	21.6	0.4	20.6	0.4	52.1	0.5
30	47.9	0.5	32.2	0.5	30.1	0.5	30.4	0.5	66.7	0.5
40	62.1	0.5	44.2	0.5	41.0	0.5	40.7	0.5	77.7	0.4
50	74.5	0.4	54.3	0.5	50.0	0.5	48.7	0.5	85.5	0.4
60	82.5	0.4	64.3	0.5	59.1	0.5	57.9	0.5	90.1	0.3
70	88.9	0.3	73.0	0.4	66.9	0.5	65.0	0.5	94.3	0.2
80	93.2	0.3	79.6	0.4	74.1	0.4	72.3	0.4	96.7	0.2
90	95.4	0.2	84.6	0.4	78.7	0.4	77.5	0.4	97.7	0.2
100	97.4	0.2	89.2	0.3	83.6	0.4	81.9	0.4	98.6	0.1

Tab. B.38. Power in the case of the 3D Clayton copula (0.5) with Exponential margins

N	mBLAW	(sd)	mHHG	(sd)	BBL	(sd)	mHSIC	(sd)	GR	(sd)
10	13.9	0.3	11.5	0.3	12.2	0.3	9.9	0.3	34.2	0.5
20	30.2	0.5	19.6	0.4	21.6	0.4	12.7	0.3	52.1	0.5
30	47.9	0.5	27.6	0.4	30.1	0.5	15.6	0.4	66.7	0.5
40	62.1	0.5	34.9	0.5	41.0	0.5	18.9	0.4	77.7	0.4
50	74.5	0.4	42.7	0.5	50.0	0.5	22.4	0.4	85.5	0.4
60	82.5	0.4	49.1	0.5	59.1	0.5	26.9	0.4	90.1	0.3
70	88.9	0.3	56.2	0.5	66.9	0.5	30.9	0.5	94.3	0.2
80	93.2	0.3	61.8	0.5	74.1	0.4	36.0	0.5	96.7	0.2
90	95.4	0.2	66.9	0.5	78.7	0.4	38.7	0.5	97.7	0.2
100	97.4	0.2	72.6	0.4	83.6	0.4	44.4	0.5	98.6	0.1

Tab. B.39. Power in the case of the 3D Clayton copula (0.5) with Cauchy margins

N	mBLAW	(sd)	mHHG	(sd)	BBL	(sd)	mHSIC	(sd)	GR	(sd)
10	11.6	0.3	10.9	0.3	9.6	0.3	9.7	0.3	23.9	0.4
20	23.3	0.4	19.8	0.4	15.6	0.4	13.6	0.3	35.1	0.5
30	34.6	0.5	29.2	0.5	21.0	0.4	18.0	0.4	47.0	0.5
40	46.4	0.5	39.3	0.5	28.3	0.5	22.8	0.4	56.7	0.5
50	57.4	0.5	48.5	0.5	34.6	0.5	28.8	0.5	66.7	0.5
60	65.6	0.5	56.8	0.5	40.9	0.5	34.2	0.5	72.8	0.4
70	74.1	0.4	65.0	0.5	48.2	0.5	39.4	0.5	79.1	0.4
80	79.0	0.4	70.1	0.5	52.4	0.5	45.1	0.5	83.7	0.4
90	84.4	0.4	76.0	0.4	59.0	0.5	51.3	0.5	87.7	0.3
100	87.8	0.3	80.9	0.4	63.5	0.5	55.0	0.5	90.1	0.3

Tab. B.40. Power in the case of the 3D Gumbel copula (1.2) with Normal margins

N	mBLAW	(sd)	mHHG	(sd)	BBL	(sd)	mHSIC	(sd)	GR	(sd)
10	11.6	0.3	14.3	0.4	9.6	0.3	11.6	0.3	23.9	0.4
20	23.3	0.4	27.2	0.4	15.6	0.4	16.4	0.4	35.1	0.5
30	34.6	0.5	39.0	0.5	21.0	0.4	21.0	0.4	47.0	0.5
40	46.4	0.5	50.1	0.5	28.3	0.5	26.5	0.4	56.7	0.5
50	57.4	0.5	60.5	0.5	34.6	0.5	32.4	0.5	66.7	0.5
60	65.6	0.5	68.1	0.5	40.9	0.5	37.9	0.5	72.8	0.4
70	74.1	0.4	75.7	0.4	48.2	0.5	43.5	0.5	79.1	0.4
80	79.0	0.4	80.7	0.4	52.4	0.5	49.3	0.5	83.7	0.4
90	84.4	0.4	85.3	0.4	59.0	0.5	54.6	0.5	87.7	0.3
100	87.8	0.3	88.5	0.3	63.5	0.5	59.3	0.5	90.1	0.3

Tab. B.41. Power in the case of the 3D Gumbel copula (1.2) with Exponential margins

N	mBLAW	(sd)	mHHG	(sd)	BBL	(sd)	mHSIC	(sd)	GR	(sd)
10	11.6	0.3	10.0	0.3	9.6	0.3	8.6	0.3	23.9	0.4
20	23.3	0.4	16.1	0.4	15.6	0.4	10.0	0.3	35.1	0.5
30	34.6	0.5	21.6	0.4	21.0	0.4	11.2	0.3	47.0	0.5
40	46.4	0.5	27.1	0.4	28.3	0.5	13.1	0.3	56.7	0.5
50	57.4	0.5	33.2	0.5	34.6	0.5	14.9	0.4	66.7	0.5
60	65.6	0.5	38.8	0.5	40.9	0.5	17.6	0.4	72.8	0.4
70	74.1	0.4	44.8	0.5	48.2	0.5	19.3	0.4	79.1	0.4
80	79.0	0.4	48.6	0.5	52.4	0.5	21.6	0.4	83.7	0.4
90	84.4	0.4	53.1	0.5	59.0	0.5	24.3	0.4	87.7	0.3
100	87.8	0.3	57.5	0.5	63.5	0.5	25.3	0.4	90.1	0.3

TAB. B.42. Power in the case of the 3D Gumbel copula (1.2) with Cauchy margins

N	mBLAW	(sd)	mHHG	(sd)	BBL	(sd)	mHSIC	(sd)	GR	(sd)
10	11.5	0.3	10.3	0.3	12.4	0.3	9.9	0.3	19.3	0.4
20	24.0	0.4	19.9	0.4	22.9	0.4	17.6	0.4	30.2	0.5
30	39.9	0.5	31.5	0.5	37.1	0.5	25.2	0.4	42.7	0.5
40	55.1	0.5	43.9	0.5	53.1	0.5	33.4	0.5	53.3	0.5
50	67.9	0.5	55.9	0.5	64.2	0.5	43.9	0.5	64.2	0.5
60	79.4	0.4	66.0	0.5	74.9	0.4	51.9	0.5	71.8	0.4
70	87.3	0.3	76.0	0.4	82.3	0.4	61.5	0.5	80.0	0.4
80	91.7	0.3	81.8	0.4	87.7	0.3	68.8	0.5	84.2	0.4
90	94.6	0.2	86.5	0.3	90.9	0.3	74.4	0.4	88.1	0.3
100	97.3	0.2	91.3	0.3	95.0	0.2	81.2	0.4	92.3	0.3

TAB. B.43. Power in the case of the 3D Normal copula ($\rho_{x,y} = 0.0, \rho_{x,z} = 0.1, \rho_{y,z} = 0.5$) with Normal margins

N	mBLAW	(sd)	mHHG	(sd)	BBL	(sd)	mHSIC	(sd)	GR	(sd)
10	11.5	0.3	11.5	0.3	12.4	0.3	11.0	0.3	19.3	0.4
20	24.0	0.4	22.7	0.4	22.9	0.4	18.2	0.4	30.2	0.5
30	39.9	0.5	36.2	0.5	37.1	0.5	27.4	0.4	42.7	0.5
40	55.1	0.5	49.8	0.5	53.1	0.5	35.2	0.5	53.3	0.5
50	67.9	0.5	62.2	0.5	64.2	0.5	46.0	0.5	64.2	0.5
60	79.4	0.4	72.8	0.4	74.9	0.4	54.2	0.5	71.8	0.4
70	87.3	0.3	81.1	0.4	82.3	0.4	63.0	0.5	80.0	0.4
80	91.7	0.3	86.5	0.3	87.7	0.3	70.2	0.5	84.2	0.4
90	94.6	0.2	90.8	0.3	90.9	0.3	74.8	0.4	88.1	0.3
100	97.3	0.2	94.2	0.2	95.0	0.2	81.8	0.4	92.3	0.3

TAB. B.44. Power in the case of the 3D Normal copula ($\rho_{x,y} = 0.0, \rho_{x,z} = 0.1, \rho_{y,z} = 0.5$) with Exponential margins

N	mBLAW	(sd)	mHHG	(sd)	BBL	(sd)	mHSIC	(sd)	GR	(sd)
10	11.5	0.3	8.9	0.3	12.4	0.3	7.3	0.3	19.3	0.4
20	24.0	0.4	14.2	0.3	22.9	0.4	10.3	0.3	30.2	0.5
30	39.9	0.5	20.0	0.4	37.1	0.5	12.0	0.3	42.7	0.5
40	55.1	0.5	27.0	0.4	53.1	0.5	14.8	0.4	53.3	0.5
50	67.9	0.5	33.8	0.5	64.2	0.5	18.1	0.4	64.2	0.5
60	79.4	0.4	40.4	0.5	74.9	0.4	21.1	0.4	71.8	0.4
70	87.3	0.3	47.8	0.5	82.3	0.4	24.9	0.4	80.0	0.4
80	91.7	0.3	53.9	0.5	87.7	0.3	28.5	0.5	84.2	0.4
90	94.6	0.2	60.0	0.5	90.9	0.3	31.1	0.5	88.1	0.3
100	97.3	0.2	65.3	0.5	95.0	0.2	35.5	0.5	92.3	0.3

TAB. B.45. Power in the case of the 3D Normal copula ($\rho_{x,y} = 0.0, \rho_{x,z} = 0.1, \rho_{y,z} = 0.5$) with Cauchy margins

B.2.2. Random vectors examples

N	mBLAW	(sd)	mHHG	(sd)	mHSIC	(sd)	BBL	(sd)	KOJA	(sd)
10	4.9	0.3	5.2	0.3	10.3	0.4	0.0	0.0	1.1	0.1
20	10.8	0.4	7.8	0.4	20.6	0.6	4.4	0.3	0.6	0.1
30	17.5	0.5	10.6	0.4	31.4	0.7	4.5	0.3	0.4	0.1
40	27.7	0.6	13.0	0.5	45.2	0.7	5.9	0.3	1.5	0.2
50	39.0	0.7	15.3	0.5	60.7	0.7	5.0	0.3	2.5	0.2
60	52.2	0.7	21.0	0.6	76.1	0.6	4.8	0.3	4.6	0.3
70	63.9	0.7	24.5	0.6	85.2	0.5	5.3	0.3	7.3	0.4

TAB. B.46. Power in the case of vectors with pairwise independent components, case mixed

N	mBLAW	(sd)	mHHG	(sd)	mHSIC	(sd)
10	5.2	0.2	5.4	0.2	6.9	0.3
20	7.9	0.3	7.3	0.3	10.1	0.3
40	15.8	0.4	10.1	0.3	18.0	0.4
60	26.9	0.4	13.4	0.3	28.6	0.5
80	43.8	0.5	17.8	0.4	42.2	0.5
100	61.6	0.5	22.7	0.4	59.1	0.5
150	93.1	0.3	38.2	0.5	88.8	0.3
200	99.8	0.0	55.8	0.5	99.0	0.1
250	100.0	0.0	73.6	0.4	100.0	0.0

TAB. B.47. Power in the case of vectors with pairwise independent components, case hidden

N	mBLAW	(sd)	mHHG	(sd)	mHSIC	(sd)
10	11.0	0.3	12.3	0.3	12.3	0.3
20	26.0	0.4	26.4	0.4	21.0	0.4
30	44.2	0.5	41.4	0.5	29.7	0.5
40	60.3	0.5	56.5	0.5	40.6	0.5
50	72.9	0.4	69.0	0.5	51.2	0.5
60	82.9	0.4	78.3	0.4	60.4	0.5
70	88.6	0.3	84.8	0.4	68.4	0.5
80	93.1	0.3	89.4	0.3	75.9	0.4
90	95.9	0.2	93.5	0.2	81.7	0.4
100	97.6	0.2	96.1	0.2	86.3	0.3

TAB. B.48. Power in the case of vectors for the Clayton (0.3) copula with Normal margins

N	mBLAW	(sd)	mHHG	(sd)	mHSIC	(sd)
10	11.0	0.3	8.6	0.3	10.7	0.3
20	26.0	0.4	13.0	0.3	16.5	0.4
30	44.2	0.5	18.1	0.4	23.4	0.4
40	60.3	0.5	24.9	0.4	30.1	0.5
50	72.9	0.4	31.4	0.5	38.1	0.5
60	82.9	0.4	38.4	0.5	44.6	0.5
70	88.6	0.3	45.9	0.5	51.9	0.5
80	93.1	0.3	52.3	0.5	58.8	0.5
90	95.9	0.2	58.9	0.5	64.0	0.5
100	97.6	0.2	64.8	0.5	69.1	0.5

Tab. B.49. Power in the case of vectors for the Clayton (0.3) copula with Exponential margins

N	mBLAW	(sd)	mHHG	(sd)	mHSIC	(sd)
10	11.0	0.3	10.5	0.3	10.0	0.3
20	26.0	0.4	16.9	0.4	12.8	0.3
30	44.2	0.5	22.7	0.4	15.9	0.4
40	60.3	0.5	30.1	0.5	19.5	0.4
50	72.9	0.4	35.9	0.5	22.6	0.4
60	82.9	0.4	41.9	0.5	27.5	0.4
70	88.6	0.3	47.3	0.5	30.6	0.5
80	93.1	0.3	51.2	0.5	34.7	0.5
90	95.9	0.2	57.6	0.5	38.3	0.5
100	97.6	0.2	62.8	0.5	43.0	0.5

Tab. B.50. Power in the case of vectors for the Clayton (0.3) copula with Cauchy margins

N	mBLAW	(sd)	mHHG	(sd)	mHSIC	(sd)
10	8.9	0.3	8.5	0.3	8.5	0.3
20	16.7	0.4	15.5	0.4	11.2	0.3
30	26.4	0.4	23.7	0.4	15.1	0.4
40	36.1	0.5	29.4	0.5	18.4	0.4
50	45.5	0.5	36.1	0.5	23.3	0.4
60	54.1	0.5	42.6	0.5	26.6	0.4
70	60.2	0.5	48.6	0.5	31.5	0.5
80	67.6	0.5	55.0	0.5	35.5	0.5
90	73.0	0.4	59.5	0.5	40.4	0.5
100	77.8	0.4	64.6	0.5	45.0	0.5

Tab. B.51. Power in the case of vectors for the Gumbel (1.1) copula with Normal margins

N	mBLAW	(sd)	mHHG	(sd)	mHSIC	(sd)
10	8.9	0.3	10.5	0.3	9.2	0.3
20	16.7	0.4	19.0	0.4	10.8	0.3
30	26.4	0.4	26.1	0.4	13.1	0.3
40	36.1	0.5	32.9	0.5	15.8	0.4
50	45.5	0.5	39.4	0.5	18.8	0.4
60	54.1	0.5	45.2	0.5	21.5	0.4
70	60.2	0.5	50.7	0.5	24.2	0.4
80	67.6	0.5	56.4	0.5	26.3	0.4
90	73.0	0.4	59.7	0.5	30.5	0.5
100	77.8	0.4	65.1	0.5	33.9	0.5

TAB. B.52. Power in the case of vectors for the Gumbel (1.1) copula with Exponential margins

N	mBLAW	(sd)	mHHG	(sd)	mHSIC	(sd)
10	8.9	0.3	7.5	0.3	7.4	0.3
20	16.7	0.4	10.2	0.3	7.3	0.3
30	26.4	0.4	13.5	0.3	8.8	0.3
40	36.1	0.5	14.8	0.4	8.6	0.3
50	45.5	0.5	18.2	0.4	9.8	0.3
60	54.1	0.5	18.7	0.4	10.2	0.3
70	60.2	0.5	21.8	0.4	11.3	0.3
80	67.6	0.5	23.8	0.4	12.1	0.3
90	73.0	0.4	26.3	0.4	13.6	0.3
100	77.8	0.4	28.1	0.4	13.7	0.3

TAB. B.53. Power in the case of vectors for the Gumbel (1.1) copula with Cauchy margins

N	mBLAW	(sd)	mHHG	(sd)	mHSIC	(sd)
10	9.0	0.3	9.3	0.3	11.3	0.3
20	21.0	0.4	19.7	0.4	20.2	0.4
30	38.7	0.5	32.6	0.5	33.4	0.5
40	53.1	0.5	44.9	0.5	43.7	0.5
50	67.5	0.5	58.0	0.5	55.5	0.5
60	79.4	0.4	69.4	0.5	67.6	0.5
70	86.8	0.3	77.2	0.4	75.7	0.4
80	92.1	0.3	84.3	0.4	83.5	0.4
90	96.0	0.2	90.1	0.3	89.0	0.3
100	97.5	0.2	93.3	0.2	92.5	0.3

TAB. B.54. Power in the case of vectors for the Normal copula ($\rho_{x,y} = \rho_{x,z} = 0.1, \rho_{y,z} = 0.3$) with Normal margins

N	mBLAW	(sd)	mHHG	(sd)	mHSIC	(sd)
10	9.0	0.3	12.3	0.3	13.9	0.3
20	21.0	0.4	24.6	0.4	24.3	0.4
30	38.7	0.5	38.2	0.5	37.4	0.5
40	53.1	0.5	50.8	0.5	48.3	0.5
50	67.5	0.5	63.0	0.5	57.9	0.5
60	79.4	0.4	73.6	0.4	68.8	0.5
70	86.8	0.3	79.9	0.4	76.0	0.4
80	92.1	0.3	86.3	0.3	82.8	0.4
90	96.0	0.2	91.5	0.3	87.8	0.3
100	97.5	0.2	94.3	0.2	91.1	0.3

Tab. B.55. Power in the case of vectors for the Normal copula ($\rho_{x,y} = \rho_{x,z} = 0.1, \rho_{y,z} = 0.3$) with Exponential margins

N	mBLAW	(sd)	mHHG	(sd)	mHSIC	(sd)
10	9.0	0.3	7.7	0.3	8.0	0.3
20	21.0	0.4	11.2	0.3	9.8	0.3
30	38.7	0.5	14.4	0.4	11.9	0.3
40	53.1	0.5	18.3	0.4	13.2	0.3
50	67.5	0.5	22.2	0.4	16.9	0.4
60	79.4	0.4	26.8	0.4	18.8	0.4
70	86.8	0.3	30.6	0.5	21.5	0.4
80	92.1	0.3	35.4	0.5	24.6	0.4
90	96.0	0.2	38.8	0.5	28.0	0.4
100	97.5	0.2	44.2	0.5	30.6	0.5

Tab. B.56. Power in the case of vectors for the Normal copula ($\rho_{x,y} = \rho_{x,z} = 0.1, \rho_{y,z} = 0.3$) with Cauchy margins

Appendix C

CODE

C.1. RCPP CODE

To compute the test statistic T in the multivariate case, we use the following RCPP functions.

```
/* Author: Guillaume Boglioni Beaulieu
 Description: RCPP functions to calculate the test
    statistic and its p-value
   'sample_mat' is a function to obtain a copy of a matrix
       with randomly permutated rows
   'rcpp_dist' computes all pairs of distances between the
       vectors (rows) of a matrix. It is the equivalent of
       the R function 'dist'
   'comp_T_RCPP' computes the test statistic T and its
      assossiated p-value
 Works for: three (3) random  variables AND three random
    vectors (possibly of different sizes)
 Last update: 10/31/2016 */


/* Function to create a copy of a matrix where the rows
   have been randomly permutated */
/* The only argument is the original matrix*/
// [[Rcpp::depends(RcppArmadillo)]]
#include <RcppArmadilloExtensions/sample.h>
using namespace Rcpp ;
```

C-ii

```
// [[Rcpp::export]]
NumericMatrix sample_mat(NumericMatrix X) {
  int i;
  int N = X.nrow(); /* Sample size*/
  int v = X.ncol(); /* Dimension of an observation */
  NumericMatrix sample_mat(N, v);

  IntegerVector N_seq = seq_len(N) - 1; /* Sequence of
     integers from 0 to N-1 */
  IntegerVector N_perm = RcppArmadillo::sample(N_seq, N,
     FALSE); /* Permutation of the integers sequence */

  /* Each row of sample_mat is randomly selected from the
     rows of X */
  for (i = 0; i < N; i++) {
    sample_mat(i, _) = X(N_perm[i],_);
  }
  return sample_mat;
/*return Rcpp::wrap(sub_X); */
}

/* Function to calculate the distances between all pairs
   of vectors (rows) in a matrix */
/* It is the equivalent of the R function dist() */
/* The only argument is the original matrix*/
// ** Found here: http://stackoverflow.com/questions
   /36829700/rcpp-my-distance-matrix-program-is-slower-
   than-the-function-in-package ** //
#include <RcppArmadillo.h>

// [[Rcpp::export]]
NumericMatrix rcpp_dist(NumericMatrix X) {
  int outrows = X.nrow(), i = 0, j = 0;
  double d;
  NumericMatrix out(outrows,outrows);
```

```
  for (i = 0; i < outrows - 1; i++){
    NumericVector v1 = X.row(i);
    for (j = i + 1; j < outrows ; j ++){
      d = sqrt(sum(pow(v1-X.row(j), 2.0)));
      out(j,i)=d;
      out(i,j)=d;
    }
  }

  return out;
}


/* Main function that computes the test statistic T and
   its assossiated p-value */
/* Takes 5 arguments: */
/*    'X', 'Y', 'Z': the three matrices of observations */
/*    'n_perm' the number of permutations on which is
   based the p-value */
/*    't_star' the critical value for T. H_0 will be
   rejected if T > t_star. */
/* Only if t_star == 0.0, the permutation method is used.
   */
/* Returns two results: */
/*    T, the test statistic */
/*    if t_star == 0, the p-value based on the
   permutations. Else, an integer (1 for rejection, 0 for
   acceptance of H0) */
#include <RcppArmadillo.h>

// [[Rcpp::export]]

NumericVector comp_T_RCPP(NumericMatrix X, NumericMatrix Y
   , NumericMatrix Z, int n_perm, double t_star = 0.0) {
```

```
/* Declare variables*/
NumericVector results(2); /* Final output containg the
    test statistic and p-value*/
int m, i, j, k; /* Count integers for the For loops */
double count = 0.0; /* Count of the number of times
   the test statistic based on a permutation sample is
    bigger than the original test statistic*/
int N = X.nrow(), vx = X.ncol(), vy = Y.ncol(), vz = Z
   .ncol(); /* Sample size and size of vectors X, Y, Z
    */
NumericMatrix dx(N, N), dy(N, N), dz(N, N); /*
   Matrices of distances between each point */
double T = 0.0; /* Test statistic*/
double S = 0.0; /* Permutated test statistic */
double S_ij; /* Component of the test statistic (based
    on the pair of point (i,j))*/
double R_x0, R_y0, R_z0; /* Radii: distances between (
   x0, y0, z0) and (xj, yj, zj) */

double A_111, A_112, A_121, A_122, A_211, A_212, A_221
   , A_222; /* Components of the test statistic*/
double Ai_1, Ai_2, Aj_1, Aj_2, Ak_1, Ak_2;


for (m = 0; m < n_perm +1 ; m++) { /* Most outer loop:
   Done for the original sample and n_perm permutation
   samples*/
  NumericMatrix sub_X(N, vx);
  NumericMatrix sub_Y(N, vy);
  NumericMatrix sub_Z(N, vz);
  S = 0.0; /* Initialize test statistic to 0 */
```

```
  /* The very first time, we don't want a permutation
     sample, but the real sample to calculate the real
     test statistic , T*/
  if (m == 0) {
    sub_X = X;
    sub_Y = Y;
    sub_Z = Z;
  }
  /* Every other time, we want a random permutation of
     the sample */
  else {
    sub_X = sample_mat(X);
    sub_Y = sample_mat(Y);
    sub_Z = sample_mat(Z);
  }

  /*Compute the matrices of distances */
  dx = rcpp_dist(sub_X);
  dy = rcpp_dist(sub_Y);
  dz = rcpp_dist(sub_Z);

/* To calculate the test statistic , we compute S(i,j)
   for every pair of points */
/* Hence, we have a double loop on i and j, with the
   restriction that i is different from j */
   for (i = 0; i < N; i++) {
     for (j = 0; j < N; j++) {
       if (i == j)
       {continue;} /* We skip cases where i == j */

       /* Reinitialize variables*/
       A_111 = 0.0;
       A_112 = 0.0;
       A_121 = 0.0;
       A_122 = 0.0;
```

```
        A_211 = 0.0;
        A_212 = 0.0;
        A_221 = 0.0;
        A_222 = 0.0;
        Ai_1 = 0.0;
        Ai_2 = 0.0;
        Aj_1 = 0.0;
        Aj_2 = 0.0;
        Ak_1 = 0.0;
        Ak_2 = 0.0;
        S_ij = 0.0;

        R_x0 = dx(i,j);
        R_y0 = dy(i,j);
        R_z0 = dz(i,j);


    /* Categorizing each data point that is not i or j
       in the 2 x 2 x 2 contingency table*/
    /* There are N-2 points not equal to i or j*/
    for ( k = 0; k < N; k++ ) {
      if ((k == i) || (k == j))
      {continue;} /* Skip cases where k = i OR j*/

      if      ((dx(i,k) <= R_x0) && (dy(i,k) <= R_y0) &&
          (dz(i,k) <= R_z0)) {A_111++;}
      else if ((dx(i,k) <= R_x0) && (dy(i,k) <= R_y0) &&
          (dz(i,k) >  R_z0)) {A_112++;}
      else if ((dx(i,k) <= R_x0) && (dy(i,k) >  R_y0) &&
          (dz(i,k) <= R_z0)) {A_121++;}
      else if ((dx(i,k) <= R_x0) && (dy(i,k) >  R_y0) &&
          (dz(i,k) >  R_z0)) {A_122++;}
      else if ((dx(i,k) >  R_x0) && (dy(i,k) <= R_y0) &&
          (dz(i,k) <= R_z0)) {A_211++;}
```

```
  else if ((dx(i,k) >  R_x0) && (dy(i,k) <= R_y0) &&
      (dz(i,k) >  R_z0)) {A_212++;}
  else if ((dx(i,k) >  R_x0) && (dy(i,k) >  R_y0) &&
      (dz(i,k) <= R_z0)) {A_221++;}
  else {A_222++;}

} /* End k loop */



/* Calculate the totals*/
Ai_1 = A_111 + A_112 + A_121 + A_122;
Ai_2 = A_211 + A_212 + A_221 + A_222;

Aj_1 = A_111 + A_112 + A_211 + A_212;
Aj_2 = A_121 + A_122 + A_221 + A_222;

Ak_1 = A_111 + A_121 + A_211 + A_221;
Ak_2 = A_112 + A_122 + A_212 + A_222;



/* Calculate S(i,j) */
if (Ai_1 == 0  || Ai_2 == 0   || Aj_1 == 0   || Aj_2 ==
    0 || Ak_1 == 0   || Ak_2 == 0 ) {
  S_ij = 0.0;
}
    else {
    S_ij = (N-2) * (N-2) *
        ((A_111 * A_111)/(double)(Ai_1 * Aj_1 * Ak_1
          ) +
         (A_112 * A_112)/(double)(Ai_1 * Aj_1 * Ak_2
           ) +
         (A_121 * A_121)/(double)(Ai_1 * Aj_2 * Ak_1
           ) +
         (A_122 * A_122)/(double)(Ai_1 * Aj_2 * Ak_2
           ) +
```

```
            (A_211 * A_211)/(double)(Ai_2 * Aj_1 * Ak_1
               ) +
            (A_212 * A_212)/(double)(Ai_2 * Aj_1 * Ak_2
               ) +
            (A_221 * A_221)/(double)(Ai_2 * Aj_2 * Ak_1
               ) +
            (A_222 * A_222)/(double)(Ai_2 * Aj_2 * Ak_2
               )) - (N-2);
        }

        /* The test statistic is the sum over all S(i,j)
           */
        S = S + S_ij;

      } /* End j loop */
    } /* End i loop */

    if (m == 0) {
      T = S; /* For the very first iteration of the most
           outer loop, the quantity S is our real test
           statistic T*/
        if (t_star != 0.0){   /* If a critical value has
            been provided, we use it to see wheter or not
             H_0 is rejected. In this case, the fonction
            will return 1 for a rejection and 0 otherwise
             */
          if (T >= t_star){
            results(1) = 1;    /* Reject H_0 */
          }
          else{
            results(1) = 0;    /* Do not reject H_0 */
          }


        }
```

```
    }

    else { /* For all other iterations, we check if S >=
        T and if it's the case add to the counter */
      if (T <= S) {
        count = count + 1;
      }
    }

  } /* End of m loop   */

  /* What to return */

    results(0) = T;

  if (t_star == 0.0){ /* If no critical values have been
      provided, we return the p-value based on the
     permutated trials */
     results(1) = (double)count / (double)n_perm;
  }

  /* return Rcpp::wrap(S); (double)count / (double)
     n_perm */
  return(results);

}      // End Cfunc
```

C.2. R CODE

To conduct the power simulations we used R code. However, note that for the multivariate examples the R function ‘power’ below calls the RCPP function ‘comp_T_RCPP’ described previously in section C.1.

```
# Author: Guillaume Boglioni Beaulieu
# Description: This program contains a function 'power' to
    compute power simulations for several independence
   tests
# In the 2D case (testing indep. between X and Y) the
   following tests of independence are implemented:
#      - Original 'Heller et al.' test (HHG)
#      - 'Heller et al.' using ranks of the observations (
   BLAW)
#      - Distance-Covariance test (DCOV)
#      - Hilbert-Smith independence criterion test (HSIC)
# In the 3D case (testing indep. between X, Y and Z) the
   following tests of independence are implemented:
#      - HHG extended to 3D (mHHG)
#      - HHG extended to 3D using ranks of the observations
     (mBLAW)
#      - Multidimentional HSIC (mHSIC)
#      - Beran-Bilodeau-Lafaye (2007) test (BBL)
#      - Genest-Remillard test (GR) (usable in 3D but only
   for random variables (1D each))
# Notes:
#      # Both 2D and 3D with are implemented within the
   same function 'power', see detailed description below
#      - For the 2D version, since we use Heller's test on
   ranks, no RCPP code is called. Instead we directly use
   the package 'HHG'
#      - Function 'power' outputs (and saves) three things:
     results in Latex syntax, results in R tables and
   graphs (pdf) of the results
# Last update: 11/17/2016
```

```
# Load required librairies
library(HHG)     # Package containing HHG test
library(energy) # Package containing distance-covariance
   test
library(dHSIC) # Package containing distance-covariance
   test
library(copula) # Package to generate data using copulas
library(IndependenceTests) # Package for Lafaye-Bilodeau-
   Beran test
library(Rcpp) # RCPP material
library(RcppArmadillo) # RCPP material

# Generate tables in Latex
library(xtable)
options(xtable.floating = FALSE)
options(xtable.timestamp = "")

#Use function 'dependogram' of package IndependenceTests (
   there is a funciton with the same name in package '
   copula')
dependogram <- IndependenceTests::dependogram


# Set path to where results and graphs should be saved
# Paths are different wheter I run on my PC or on the DMS
   machines
if(.Platform$OS.type == "unix") {
  setwd("/home/boglionibe/Memoire")
  data.direct   <- "/home/boglionibe/Memoire/
     Results_copulas/"
  graphs.direct <- "/home/boglionibe/Memoire/
     Graphs_copulas/"
```

```
} else {
  setwd("C:/Seafile/Boglioni/RCPP_test")
  data.direct   <- "C:/Seafile/Boglioni/Thesis/
     Results_copulas/"
  graphs.direct <- "C:/Seafile/Boglioni/Thesis/
     Graphs_copulas/"
}

# Source the location of RCPP code (need in the 3D cases)
sourceCpp("RCPP_test.cpp", verbose = TRUE)



# Function 'power' returns the empirical power of the
   indepedence tests based on 'B' simulations (with their
   standard deviation)
# Empirical quantiles are used to make the decision on
   rejecting or not rejecting H_0
# Those empirical quantiles are based on 'M' simulations
   under H_0
# Arguments of function 'power'
#   nbr.rv          Number of random variables (or vectors)
    on which the independence tests are applied (2 or 3).
#   N:              A vector containing all sample sizes
   for which the simulations are to be done
#   B:              Number of trials for the computation of
    power
#   M:              Number of trials to approximate the
   distribtuion of T|H_0
#   alpha:          Level of the test
#   dep.type:       Structure of dependence (i.e. a
   specific H_1)
#   indep:          TRUE to generate independent X and Y (
   this is used, for instance, to check that the level of
   the test is alpha)
```

```
#   cop.para:       Parameter of the copula structure, if a
    copula structure is used to generate data
#   marginal:       Marginals of X, Y and Z, if a copula
   structure is used to generate data
#   v.dim:          Dimension of the vectors X, Y and Z, if
    a copula structure is used (hence X, Y, Z always have
   the same dimension in the copulat examples)
#   data.direct:    Directory where the power results are
   to be saved
#   graphs.direct: Directory where the graphs are to be
   saved
#   table.caption: Caption for the produced Latex table
#   data.name:      Name of the data.frame containing the
   power results
#   legend.pos:     Position of the legend on the produced
   graphs ("topleft", ...)
# Outputs of function 'power':
#   - Table of results in Latex syntax (in the R console)
#   - data.frame of results (saved in the specified
   directory)
#   - graph of results (saved in the specified director)


power <- function(nbr.rv = 2,
                  N = c(10, 20, 30, 40, 50, 60, 70, 80,
                     90, 100),
                  B = 10000,
                  M = 50000,
                  alpha = 0.05,
                  dep.type,
                  indep = FALSE,
                  cop.param = 1,
                  v.dim = 1,
                  marginal = qnorm,
```

```
                      data.direct = 'C:/Seafile/Boglioni/
                         Thesis/Results/',
                      graphs.direct = 'C:/Seafile/Boglioni/
                         Thesis/Graphs/',
                      table.caption,
                      data.name,
                      legend.pos = 'topleft'){

  # Number of samples sizes for which we run the
     simulation, 10 by default for sample sizes N = 10,
     20,..., 100
  Nbr.runs <- length(N)

  # Initialize a matrix to contain all results (one row
     for each N)
  data <- NULL

  # Beginning of outer loop. Everything is re-done for
     each sample size, so Nbr.runs times in total
  for(k in 1: Nbr.runs){

    # Initialize to zero the number of times H_0 is
       rejected (i.e. a 'success'), for each test
    # Initialize the vector to contain the sample of T|H_0
       , for each test

    # Different tests are considered in the bivariate (nbr
       .rv = 2) and trivariate (nbr.rv = 3) cases
    if(nbr.rv == 2){
      n.success.BLAW <- 0
      n.success.HHG  <- 0
      n.success.DCOV <- 0
      n.success.HSIC <- 0

      T0.BLAW <- vector(length = M)
```

```
  T0.HHG  <- vector(length = M)
  T0.DCOV <- vector(length = M)
  T0.HSIC <- vector(length = M)
}
else{
  n.success.mBLAW <- 0
  n.success.mHHG  <- 0
  n.success.mHSIC  <- 0
  n.success.BBL   <- 0
  n.success.GR    <- 0

  T0.mBLAW <- vector(length = M)
  T0.mHHG  <- vector(length = M)
  T0.mHSIC  <- vector(length = M)
  T0.BBL   <- vector(length = M)
  T0.GR    <- vector(length = M)
}

# Initialize to NULL the Vector to contain power
   restults (for one specific N)
results <- NULL

# Beginning of 'For' loop: 'M' iterations to have a
   sample of T|H_0, 'B' iterations to estimate the
   power

for (i in 1: (M + B)) {

  # Generate data according to the specified
     dependence structure (independent data for the
     first 'M' iterations)

  ### 2D CASES ###

  # 4 independent clouds
```

```
if(dep.type == 'indep.clouds'){
  X <- matrix(sample( c(-1,1), size = N[k], replace
     = TRUE ) + rnorm(N[k])/3, nrow = N[k], ncol=1)
  Y <- matrix(sample( c(-1,1), size = N[k], replace
     = TRUE ) + rnorm(N[k])/3, nrow = N[k], ncol=1)
}

# W-shape
if(dep.type == 'w.shape'){
  xu <- runif(N[k], -1, 1)
  X  <- matrix(xu + runif(N[k])/3, nrow = N[k], ncol
     =1)

  # Y under the null
  if((i <= M) || (indep == TRUE)){
    yu <- runif(N[k], -1, 1)
    Y <- matrix(4*( ( yu^2 - 1/2 )^2 + runif(N[k])/N
        [k]), nrow = N[k], ncol=1)
  }
  else{
    # Y under H_1 (to obtain T)
    Y <- matrix(4*( ( xu^2 - 1/2 )^2 + runif(N[k])/N
        [k]), nrow = N[k], ncol=1)
  }
}

# Diamond
if(dep.type == 'diamond'){

  theta <- -pi/4
  rr <- rbind( c(cos(theta), -sin(theta) ),
               c( sin(theta), cos(theta) ) )

  x1 <- runif(N[k], -1, 1 )
  y1 <- runif(N[k], -1, 1 )
```

```
  tmp.dep <- cbind( x1, y1 ) %*% rr
  X <- matrix(tmp.dep[,1], nrow = N[k], ncol=1)

  # We genreate Y under the null
  if((i <= M) || (indep == TRUE)){
    x2 <- runif(N[k], -1, 1 )
    y2 <- runif(N[k], -1, 1 )
    tmp.ind <- cbind( x2, y2 ) %*% rr
    Y <- matrix(tmp.ind[,2], nrow = N[k], ncol=1)
  }
  else{
    # We generate Y under H_1 to obtain T
    Y <- matrix(tmp.dep[,2], nrow = N[k], ncol=1)
  }
}

# Parabola
if(dep.type == 'parabola'){
  X <- matrix(runif(N[k], -1, 1), nrow = N[k], ncol
     =1)

  # We genreate Y under the null
  if((i <= M) || (indep == TRUE)){
    yu <- runif(N[k], -1, 1)
    Y <- matrix((yu^2 + runif(N[k]))/2, nrow = N[k],
         ncol=1)
  }
  else{
    # We generate Y under H_1 to obtain T
    Y <- matrix((X^2 + runif(N[k]))/2, nrow = N[k],
        ncol=1)
  }
}

# 2 - Parabolas
```

```
if(dep.type == 'two.parabolas'){
  X <- matrix(runif(N[k], -1, 1), nrow = N[k], ncol
     =1)

  # We generate Y under the null
  if((i <= M) || (indep == TRUE)){
    yu <- runif(N[k], -1, 1)
    Y <- matrix((yu^2 + runif(N[k])/2 )*( sample( c
        (-1,1), size=N[k], replace = TRUE ) ), nrow =
         N[k], ncol=1)
  }
  else{
    # We generate Y under H_1 to obtain T
    Y <- matrix((X^2 + runif(N[k])/2 )*( sample( c
        (-1,1), size=N[k], replace = TRUE ) ), nrow =
         N[k], ncol=1)
  }
}

# Circle
if(dep.type == 'circle'){
  xu <- runif(N[k], -1, 1)
  X <- matrix(sin( xu*pi ) + rnorm( N[k] )/8, nrow =
      N[k], ncol=1)

  # We generate Y under the null
  if((i <= M) || (indep == TRUE)){
    yu <- runif(N[k], -1, 1)
    Y <- matrix(cos( yu*pi ) + rnorm( N[k] )/8, nrow
         = N[k], ncol=1)
  }
  else{
    # We generate Y under H_1 to obtain T
    Y <- matrix(cos( xu*pi ) + rnorm( N[k] )/8, nrow
         = N[k], ncol=1)
```

```
    }
  }

  # Linear
  if(dep.type == 'linear'){
    X <- matrix(runif(N[k]), nrow = N[k], ncol=1)

    # We generate Y under the null
    if((i <= M) || (indep == TRUE)){
      yu <- runif(N[k])
      Y <- matrix(yu + rnorm(N[k], 0, 0.5), nrow = N[k
          ], ncol=1)
    }
    else{
      # We genreate X,Y,Z under H_1 to obtain T
      Y <- matrix(X + rnorm(N[k], 0, 0.5), nrow = N[k
          ], ncol=1)
    }
  }

  # Exponential
  if(dep.type == 'exponential'){
    X <- matrix(runif(N[k], -3, 3), nrow = N[k], ncol
        =1)

    # We generate Y under the null
    if((i <= M) || (indep == TRUE)){
      yu <- runif(N[k], -3, 3)
      Y <- matrix(exp(yu/3) + runif(N[k], -3, 3), nrow
          = N[k], ncol=1)
    }
    else{
      # We genreate X,Y,Z under H_1 to obtain T
      Y <- matrix(exp(X/3) + runif(N[k], -3, 3), nrow
          = N[k], ncol=1)
```

```
    }
  }

  # Sine
  if(dep.type == 'sine'){
    X <- matrix(runif(N[k], 0, 2*pi), nrow = N[k],
        ncol=1)

    # We generate Y under the null
    if((i <= M) || (indep == TRUE)){
      yu <- runif(N[k], 0, 2*pi)
      Y <- matrix(sin(yu) + rnorm(N[k]), nrow = N[k],
          ncol=1)
    }
    else{
      # We genreate X,Y,Z under H_1 to obtain T
      Y <- matrix(sin(X) + rnorm(N[k]), nrow = N[k],
          ncol=1)
    }
  }

  # Polynomial
  if(dep.type == 'polynomial'){
    X       <- matrix(rnorm(2*N[k]), nrow = N[k], ncol
        =2)
    epsilon <- matrix(rnorm(2*N[k], 0, 6), nrow = N[k
        ], ncol=2)

    # We generate Y under the null
    if((i <= M) || (indep == TRUE)){
      xu <- matrix(rnorm(2*N[k]), nrow = N[k], ncol=2)
      Y <-  matrix(xu + 4*xu^2 + epsilon, nrow = N[k],
          ncol=2)
    }
    else{
```

```
      # We genreate X,Y,Z under H_1 to obtain T
      Y <- matrix(X + 4*X^2 + epsilon, nrow = N[k],
          ncol=1)
    }
  }



  if(dep.type == '2d.pairwise.indep'){
    X  <- matrix(rnorm(N[k]), nrow = N[k], ncol=1)
    W  <- matrix(rnorm(N[k]), nrow = N[k], ncol=1)
    Z0 <- matrix(rnorm(N[k]), nrow = N[k], ncol=1)

    # We genreate X,Y,Z under the null
    if((i <= M) || (indep == TRUE)){
      Y <- matrix(cbind(W, Z0), nrow = N[k], ncol=2)
    }
    else{
      # We generate Y under H_1 to obtain T
      Z <- abs(Z0) * sign(X * W)
      Y <- matrix(cbind(W, Z), nrow = N[k], ncol=2)
    }
  }


  #Generated via COPULAS

  if(dep.type == '2d.clayton'){

    # We generate X,Y under the null
    if((i <= M) || (indep == TRUE)){
      data.cop.x <- rCopula(N[k], archmCopula("clayton
          ", param = cop.param, dim = 2*v.dim, use.
          indepC = "TRUE"))
```

```
      data.cop.y <- rCopula(N[k], archmCopula("clayton
          ", param = cop.param, dim = 2*v.dim, use.
          indepC = "TRUE"))
      X <- matrix(marginal(data.cop.x[, seq(1:v.dim)])
                  , nrow = N[k], ncol = v.dim)
      Y <- matrix(marginal(data.cop.y[, v.dim + seq(1:
          v.dim)]), nrow = N[k], ncol = v.dim)

      }
    else{
      # We generate X,Y under H_1 to obtain T
      data.cop <- rCopula(N[k], archmCopula("clayton",
           param = cop.param, dim = 2*v.dim, use.indepC
           = "TRUE"))
      X <- matrix(marginal(data.cop[, seq(1:v.dim)])
                 , nrow = N[k], ncol = v.dim)
      Y <- matrix(marginal(data.cop[, v.dim + seq(1:v.
          dim)]), nrow = N[k], ncol = v.dim)
    }
  }

  if(dep.type == '2d.gumbel'){

    # We generate X,Y under the null
    if((i <= M) || (indep == TRUE)){
      data.cop.x <- rCopula(N[k], archmCopula("gumbel
          ", param = cop.param, dim = 2*v.dim, use.
          indepC = "TRUE"))
      data.cop.y <- rCopula(N[k], archmCopula("gumbel
          ", param = cop.param, dim = 2*v.dim, use.
          indepC = "TRUE"))
      X <- matrix(marginal(data.cop.x[, seq(1:v.dim)])
                  , nrow = N[k], ncol = v.dim)
      Y <- matrix(marginal(data.cop.y[, v.dim + seq(1:
          v.dim)]), nrow = N[k], ncol = v.dim)
```

```
    }
    else{
      # We generate X,Y under H_1 to obtain T
      data.cop <- rCopula(N[k], archmCopula("gumbel",
          param = cop.param, dim = 2*v.dim, use.indepC
          = "TRUE"))
      X <- matrix(marginal(data.cop[, seq(1:v.dim)])
                  , nrow = N[k], ncol = v.dim)
      Y <- matrix(marginal(data.cop[, v.dim + seq(1:v.
          dim)]), nrow = N[k], ncol = v.dim)
    }
  }

  if(dep.type == '2d.normal'){

    # We generate X,Y under the null
    if((i <= M) || (indep == TRUE)){
      data.cop.x <- rCopula(N[k], normalCopula(param =
            cop.param, dim = 2*v.dim, dispstr = "un"))
      data.cop.y <- rCopula(N[k], normalCopula(param =
            cop.param, dim = 2*v.dim, dispstr = "un"))
      X <- matrix(marginal(data.cop.x[, seq(1:v.dim)])
                  , nrow = N[k], ncol = v.dim)
      Y <- matrix(marginal(data.cop.y[, v.dim + seq(1:
          v.dim)]), nrow = N[k], ncol = v.dim)

    }
    else{
      # We generate X,Y under H_1 to obtain T
      data.cop <- rCopula(N[k], normalCopula(param =
          cop.param, dim = 2*v.dim, dispstr = "un"))
      X <- matrix(marginal(data.cop[, seq(1:v.dim)])
                  , nrow = N[k], ncol = v.dim)
```

```
      Y <- matrix(marginal(data.cop[, v.dim + seq(1:v.
          dim)]), nrow = N[k], ncol = v.dim)
    }
  }



  # X and Y in 5 dimensions with, Y = log(X^2) for
     each dimension
  if(dep.type == '2d.log'){
    X <- matrix(rnorm(5*N[k]), nrow = N[k])

    # We generate Y under the null
    if((i <= M) || (indep == TRUE)){
      xu <- matrix(rnorm(5*N[k]), nrow = N[k])
      Y <-  matrix(log(xu^2), nrow = N[k])
    }
    else{
      # We generate Y under H_1 to obtain T
      Y <- matrix(log(X^2), nrow = N[k])
    }
  }

  # Y = epsilon * X
  if(dep.type == '2d.epsilon'){
    X <- matrix(rnorm(5*N[k]), nrow = N[k])

    # We generate Y under the null
    if((i <= M) || (indep == TRUE)){
      xu <- matrix(rnorm(5*N[k]), nrow = N[k])
      Y <-  matrix(xu * matrix(rnorm(5*N[k]), nrow = N
          [k]), nrow = N[k])
    }
    else{
      # We generate Y under H_1 to obtain T
```

```
      Y <-  matrix(X  * matrix(rnorm(5*N[k]), nrow = N
          [k]), nrow = N[k])
    }
  }

  # Y = B_1*X + B_2*X^2
  if(dep.type == '2d.beta'){
    X <- matrix(rnorm(5*N[k]), nrow = N[k])
    epsilon <- matrix(rnorm(5*N[k], 0, 3), nrow = N[k
        ])

    # We generate Y under the null
    if((i <= M) || (indep == TRUE)){
      xu <- matrix(rnorm(2*N[k]), nrow = N[k])
      Y  <-  matrix(cbind(xu[,1] + 4*xu[,1]^2 +
          epsilon[,1], xu[,2] + 4*xu[,2]^2 + epsilon
          [,2], epsilon[, 2 + seq(1:3)]), nrow = N[k])
    }
    else{
      # We generate Y under H_1 to obtain T
      Y <-  matrix(cbind(X[,1] + 4*X[,1]^2 +epsilon
          [,1], X[,2] + 4*X[,2]^2 + epsilon[,2],
          epsilon[, 2 + seq(1:3)]), nrow = N[k])
    }
  }


  if(dep.type == '2d.big.noise'){
    X        <- matrix(cbind(rnorm(N[k]), rnorm(N[k],
        0, 4)), nrow = N[k])
    epsilon <- matrix(rnorm(N[k], 0, 2), nrow = N[k])

    # We generate Y under the null
    if((i <= M) || (indep == TRUE)){
```

```
    xu  <- matrix(cbind(rnorm(N[k]), rnorm(N[k], 0,
        4)), nrow = N[k])
    Y  <-  matrix(xu[,1]^2 + epsilon, nrow = N[k])
  }
  else{
    # We generate Y under H_1 to obtain T
    Y  <-  matrix(X[,1]^2 + epsilon, nrow = N[k])
  }
}



### 3D CASES ###

# 3 pairwise independent N(0, 1) (still jointly
   dependent)
if(dep.type == '3d.pairwise.indep'){
  X <- matrix(rnorm(N[k]), nrow = N[k], ncol=1)
  Y <- matrix(rnorm(N[k]), nrow = N[k], ncol=1)

  # We genreate X,Y,Z under the null
  if((i <= M) || (indep == TRUE)){
    Z <- matrix(rnorm(N[k]), nrow = N[k], ncol=1)
  }
  else{
    # We generate X,Y,Z under H_1 to obtain T
    Z <- matrix(abs(rnorm(N[k])) * sign(X*Y), nrow =
        N[k], ncol=1)
  }
}

if(dep.type == '3d.cos.sin'){
  X <- matrix(rnorm(N[k], 0, 3 ), nrow = N[k], ncol
     =1)
```

```
    Y <- matrix(rnorm(N[k], 0, 3), nrow = N[k], ncol
        =1)

    # We genreate X,Y,Z under the null
    if((i <= M) || (indep == TRUE)){
      U <- rnorm(N[k], 0, 3)
      V <- rnorm(N[k], 0, 3)
      Z <- matrix(cos(U) + sin(V) + rnorm(N[k]), nrow
          = N[k], ncol=1)
    }
    else{
      # We generate X,Y,Z under H_1 to obtain T
      Z <- matrix(cos(X) + sin(Y) + rnorm(N[k]), nrow
          = N[k], ncol=1)
    }
  }

  if(dep.type == '3d.cos.exp'){
    X <- matrix(rnorm(N[k], 0, 3 ), nrow = N[k], ncol
        =1)
    Y <- matrix(rnorm(N[k], 0, 3), nrow = N[k], ncol
        =1)

    # We genreate X,Y,Z under the null
    if((i <= M) || (indep == TRUE)){
      U <- rnorm(N[k], 0, 3)
      V <- rnorm(N[k], 0, 3)
      Z <- matrix(cos(U) + exp(V/5) + rnorm(N[k]),
          nrow = N[k], ncol=1)
    }
    else{
      # We generate X,Y,Z under H_1 to obtain T
      Z <- matrix(cos(X) + exp(Y/5) + rnorm(N[k]),
          nrow = N[k], ncol=1)
    }
```

```
    }

    if(dep.type == '3d.linear'){
      X <- matrix(rnorm(N[k]), nrow = N[k], ncol=1)
      Y <- matrix(rnorm(N[k]), nrow = N[k], ncol=1)

      # We genreate X,Y,Z under the null
      if((i <= M) || (indep == TRUE)){
        U <- rnorm(N[k])
        V <- rnorm(N[k])
        Z <- matrix(U + V + rnorm(N[k], 0, 3), nrow = N[
            k], ncol=1)
      }
      else{
        # We generate X,Y,Z under H_1 to obtain T
        Z <- matrix(X + Y + rnorm(N[k], 0, 3), nrow = N[
            k], ncol=1)
      }
    }

    if(dep.type == '3d.vect.pairwise.indep'){
      X <- matrix(rnorm(2*N[k]), nrow = N[k], ncol=2)
      Z <- matrix(rnorm(2*N[k]), nrow = N[k], ncol=2)


      # We genreate X,Y,Z under the null
      if((i <= M) || (indep == TRUE)){
        Y <- matrix(rnorm(2*N[k]), nrow = N[k], ncol=2)
      }
      else{
        # We generate X,Y,Z under H_1 to obtain T
        Y <- matrix(cbind(abs(rnorm(N[k]))*sign(X[,1]*X
            [,2]), rnorm(N[k])), nrow = N[k], ncol=2)
      }
    }
```

```
if(dep.type == '3d.vect.pairwise.indep.v2'){
  X <- matrix(rnorm(2*N[k]), nrow = N[k], ncol=2)

  # We genreate X,Y,Z under the null
  if((i <= M) || (indep == TRUE)){
    Y <- matrix(rnorm(2*N[k]), nrow = N[k], ncol=2)
    Z <- matrix(rnorm(2*N[k]), nrow = N[k], ncol=2)
  }
  else{
    # We generate X,Y,Z under H_1 to obtain T
    Y1 <- rnorm(N[k])
    Z1 <- rnorm(N[k])
    Y2 <- abs(rnorm(N[k]))*sign(X[,1]*Z1)
    Z2 <- abs(rnorm(N[k]))*sign(X[,2]*Y1)

    Y <- matrix(cbind(Y1,Y2), nrow = N[k], ncol=2)
    Z <- matrix(cbind(Z1,Z2), nrow = N[k], ncol=2)
  }
}



#Generated via COPULAS

if(dep.type == '3d.clayton'){

  # We generate X,Y,Z under the null
  if((i <= M) || (indep == TRUE)){
    data.cop.x <- rCopula(N[k], archmCopula("clayton
        ", param = cop.param, dim = 3*v.dim, use.
        indepC = "TRUE"))
    data.cop.y <- rCopula(N[k], archmCopula("clayton
        ", param = cop.param, dim = 3*v.dim, use.
        indepC = "TRUE"))
```

```
      data.cop.z <- rCopula(N[k], archmCopula("clayton
         ", param = cop.param, dim = 3*v.dim, use.
         indepC = "TRUE"))
      X <- matrix(marginal(data.cop.x[, seq(1:v.dim)])
                  , nrow = N[k], ncol = v.dim)
      Y <- matrix(marginal(data.cop.y[, v.dim + seq(1:
         v.dim)])    , nrow = N[k], ncol = v.dim)
      Z <- matrix(marginal(data.cop.z[, 2*v.dim + seq
         (1:v.dim)]), nrow = N[k], ncol = v.dim)
    }
    else{
      # We generate X,Y,Z under H_1 to obtain T
      data.cop <- rCopula(N[k], archmCopula("clayton",
          param = cop.param, dim = 3*v.dim, use.indepC
          = "TRUE"))
      X <- matrix(marginal(data.cop[, seq(1:v.dim)])
                  , nrow = N[k], ncol = v.dim)
      Y <- matrix(marginal(data.cop[, v.dim + seq(1:v.
         dim)])    , nrow = N[k], ncol = v.dim)
      Z <- matrix(marginal(data.cop[, 2*v.dim + seq(1:
         v.dim)]), nrow = N[k], ncol = v.dim)
    }
  }



  if(dep.type == '3d.gumbel'){

    # We generate X,Y,Z under the null
    if((i <= M) || (indep == TRUE)){
      data.cop.x <- rCopula(N[k], archmCopula("gumbel
         ", param = cop.param, dim = 3*v.dim, use.
         indepC = "TRUE"))
```

```
      data.cop.y <- rCopula(N[k], archmCopula("gumbel
         ", param = cop.param, dim = 3*v.dim, use.
         indepC = "TRUE"))
      data.cop.z <- rCopula(N[k], archmCopula("gumbel
         ", param = cop.param, dim = 3*v.dim, use.
         indepC = "TRUE"))
      X <- matrix(marginal(data.cop.x[, seq(1:v.dim)])
                     , nrow = N[k], ncol = v.dim)
      Y <- matrix(marginal(data.cop.y[, v.dim + seq(1:
         v.dim)])    , nrow = N[k], ncol = v.dim)
      Z <- matrix(marginal(data.cop.z[, 2*v.dim + seq
         (1:v.dim)]), nrow = N[k], ncol = v.dim)
    }
    else{
      # We generate X,Y,Z under H_1 to obtain T
      data.cop <- rCopula(N[k], archmCopula("gumbel",
         param = cop.param, dim = 3*v.dim, use.indepC
         = "TRUE"))
      X <- matrix(marginal(data.cop[, seq(1:v.dim)])
                     , nrow = N[k], ncol = v.dim)
      Y <- matrix(marginal(data.cop[, v.dim + seq(1:v.
         dim)])    , nrow = N[k], ncol = v.dim)
      Z <- matrix(marginal(data.cop[, 2*v.dim + seq(1:
         v.dim)]), nrow = N[k], ncol = v.dim)
    }
  }



  if(dep.type == '3d.normal'){

    # We generate X,Y,Z under the null
    if((i <= M) || (indep == TRUE)){
      data.cop.x <- rCopula(N[k], normalCopula(param =
          cop.param, dim = 3*v.dim, dispstr = "un"))
```

```
      data.cop.y <- rCopula(N[k], normalCopula(param =
          cop.param, dim = 3*v.dim, dispstr = "un"))
      data.cop.z <- rCopula(N[k], normalCopula(param =
          cop.param, dim = 3*v.dim, dispstr = "un"))
      X <- matrix(marginal(data.cop.x[, seq(1:v.dim)])
                  , nrow = N[k], ncol = v.dim)
      Y <- matrix(marginal(data.cop.y[, v.dim + seq(1:
          v.dim)])   , nrow = N[k], ncol = v.dim)
      Z <- matrix(marginal(data.cop.z[, 2*v.dim + seq
          (1:v.dim)]), nrow = N[k], ncol = v.dim)
    }
    else{
      # We generate X,Y,Z under H_1 to obtain T
      data.cop <- rCopula(N[k], normalCopula(param =
          cop.param, dim = 3*v.dim, dispstr = "un"))
      X <- matrix(marginal(data.cop[, seq(1:v.dim)])
                  , nrow = N[k], ncol = v.dim)
      Y <- matrix(marginal(data.cop[, v.dim + seq(1:v.
          dim)])   , nrow = N[k], ncol = v.dim)
      Z <- matrix(marginal(data.cop[, 2*v.dim + seq(1:
          v.dim)]), nrow = N[k], ncol = v.dim)
    }
  }


  # Compute ranks of X and Y (and Z if applicable)
  # Size of X (number of components of the random
    vector X, hence 1 if X is a random variable)
  p <- ncol(X)
  # Size of Y
  q <- ncol(Y)
  # Initialize the matrices containg the ranks of X
    and Y
  X.rank <- NULL
  Y.rank <- NULL
```

```
# Per column ranks of X
for (j in 1:p){
  X.rank <- cbind(X.rank, rank(X[,j]))
}
# Per column ranks of Y
for (j in 1:q){
  Y.rank <- cbind(Y.rank, rank(Y[,j]))
}

# Samething for Z, if there is a Z
if(nbr.rv == 3){
  r <- ncol(Z)
  Z.rank <- NULL
  for (j in 1:r){
    Z.rank <- cbind(Z.rank, rank(Z[,j]))
  }
}

# Compute distances between each elements of X and Y
    , as well as X.rank and Y.rank (only for 2D cases
    )
# This is necessary to run the HHG test via the HHG
    package
if(nbr.rv == 2){
  Dx = as.matrix(dist((X), diag = TRUE, upper = TRUE
      ))
  Dy = as.matrix(dist((Y), diag = TRUE, upper = TRUE
      ))
  Dx.rank = as.matrix(dist((X.rank), diag = TRUE,
      upper = TRUE))
  Dy.rank = as.matrix(dist((Y.rank), diag = TRUE,
      upper = TRUE))
}
```

```
# For the first M iterations , T0 is calculated and
    we store its value
# This is done for each test (diffenrent tests in
    the 2D and 3D cases)
if(i <= M){
  if(nbr.rv == 2){
    T0.BLAW[i] <- hhg.test(Dx.rank, Dy.rank, nr.perm
        = 0)$sum.chisq
    T0.HHG[i]  <- hhg.test(Dx, Dy, nr.perm = 0)$sum.
        chisq
    T0.DCOV[i] <- dcov.test(X, Y, R = 0)$estimate
    T0.HSIC[i] <- dhsic(X, Y)$dHSIC
  }
  else{
    T0.mBLAW[i] <- comp_T_RCPP(X.rank, Y.rank, Z.
        rank, 0, 0)[1]
    T0.mHHG[i]  <- comp_T_RCPP(X, Y, Z, 0, 0)[1]
    T0.mHSIC[i] <- dhsic.test(list(X, Y, Z), B = 0)
        $statistic
    if(v.dim == 1){
      T0.GR[i]     <- indepTest(cbind(X, Y, Z), d =
          indepTestSim(n = N[k], p = 3, m = 3, N = 6,
            verbose = FALSE))$global.statistic
      T0.BBL[i]    <- dependogram(cbind(X, Y, Z), B =
            0, vecd.ou.p = c(v.dim, v.dim, v.dim),
          display = FALSE, graphics=FALSE)$Rn
    }
  }
} # End case i <= M

#Calculate the (1-alpha)-quantile of T0
if(i == M + 1){
  if(nbr.rv == 2){
    quant.BLAW <- quantile(T0.BLAW, 1 - alpha)
    quant.HHG  <- quantile(T0.HHG,  1 - alpha)
```

```
      quant.DCOV <- quantile(T0.DCOV, 1 - alpha)
      quant.HSIC <- quantile(T0.HSIC, 1 - alpha)
    }
    else{
      quant.mBLAW <- quantile(T0.mBLAW, 1 - alpha)
      quant.mHHG  <- quantile(T0.mHHG,  1 - alpha)
      quant.mHSIC <- quantile(T0.mHSIC, 1 - alpha)
      if(v.dim == 1){
        quant.GR    <- quantile(T0.GR,     1 - alpha)
        quant.BBL   <- quantile(T0.BBL,    1 - alpha)
      }
    }
  } # End case i == M + 1

  #For the remaining B iterations, we compute T for
     each test and count the number of times H0 is
     rejected in each case
  if(i >= M + 1){
    if(nbr.rv == 2){
      T.BLAW <- hhg.test(Dx.rank, Dy.rank, nr.perm =
         0)$sum.chisq
      T.HHG  <- hhg.test(Dx, Dy, nr.perm = 0)$sum.
         chisq
      T.DCOV <- dcov.test(X, Y, R = 0)$estimate
      T.HSIC <- dhsic(X, Y)$dHSIC

        if(T.BLAW > quant.BLAW) {
          n.success.BLAW = n.success.BLAW + 1
        }
        if(T.HHG  > quant.HHG) {
          n.success.HHG = n.success.HHG + 1
        }
        if(T.DCOV > quant.DCOV) {
          n.success.DCOV = n.success.DCOV + 1
        }
```

```
      if(T.HSIC > quant.HSIC) {
        n.success.HSIC = n.success.HSIC + 1
      }
  } #End case nbr.rv == 2

  else{

    if(v.dim == 1){
      T.GR     <- indepTest(cbind(X, Y, Z), d =
          indepTestSim(n = N[k], p = 3, m = 3, N = 6,
           verbose = FALSE))$global.statistic
      T.BBL    <- dependogram(cbind(X, Y, Z), B = 0,
          vecd.ou.p = c(v.dim, v.dim, v.dim), display
          = FALSE, graphics=FALSE)$Rn

      if(T.BBL > quant.BBL) {
        n.success.BBL = n.success.BBL + 1
      }
      if(T.GR > quant.GR) {
        n.success.GR = n.success.GR + 1
      }
    }

    if(comp_T_RCPP(X.rank, Y.rank, Z.rank, 0, quant.
       mBLAW)[2] == 1) {
      n.success.mBLAW = n.success.mBLAW + 1
    }
    if(comp_T_RCPP(X, Y, Z, 0, quant.mHHG)[2] == 1)
       {
      n.success.mHHG = n.success.mHHG + 1
    }

    T.mHSIC <- dhsic.test(list(X, Y, Z), B = 0)
       $statistic
    if(T.mHSIC > quant.mHSIC) {
```

```
        n.success.mHSIC = n.success.mHSIC + 1
      }

    } # End case nbr.rv == 3

  } # End case i >= M + 1


} # End for loop in i (done M+B times)

#Calculate the power based on the n.sim simulations
if(nbr.rv == 2){
  power.BLAW <- n.success.BLAW / B
  power.HHG  <- n.success.HHG  / B
  power.DCOV <- n.success.DCOV / B
  power.HSIC <- n.success.HSIC / B

  # For one sample size, place the power results in a
     vector
  results <- c(round(N[k], 0),
                   round(100*power.BLAW,1),
                   round(100*sqrt(power.BLAW * (1-power
                       .BLAW)/B), 1),
                   round(100*power.HHG, 1),
                   round(100*sqrt(power.HHG * (1-power.
                      HHG)/B), 1),
                   round(100*power.DCOV, 1),
                   round(100*sqrt(power.DCOV * (1-power
                       .DCOV)/B), 1),
                   round(100*power.HSIC, 1),
                   round(100*sqrt(power.HSIC * (1-power
                       .HSIC)/B), 1)
                )
}
else {
```

```
power.mBLAW <- n.success.mBLAW / B
power.mHHG  <- n.success.mHHG  / B
power.mHSIC <- n.success.mHSIC / B
power.BBL   <- n.success.BBL   / B
power.GR    <- n.success.GR    / B

# For one sample size, place the power results in a
   vector
if (v.dim == 1){
  results <- c(round(N[k], 0),
               round(100*power.mBLAW,1),
               round(100*sqrt(power.mBLAW * (1-power
                  .mBLAW)/B), 1),
               round(100*power.mHHG, 1),
               round(100*sqrt(power.mHHG * (1-power.
                  mHHG)/B), 1),
               round(100*power.BBL, 1),
               round(100*sqrt(power.BBL * (1-power.
                  BBL)/B), 1),
               round(100*power.mHSIC, 1),
               round(100*sqrt(power.mHSIC * (1-power
                  .mHSIC)/B), 1),
               round(100*power.GR, 1),
               round(100*sqrt(power.GR * (1-power.GR
                  )/B), 1)
              )
}
 else {
  results <- c(round(N[k], 0),
               round(100*power.mBLAW,1),
               round(100*sqrt(power.mBLAW * (1-power
                  .mBLAW)/B), 1),
               round(100*power.mHHG, 1),
               round(100*sqrt(power.mHHG * (1-power.
                  mHHG)/B), 1),
```

```
                    round(100*power.mHSIC, 1),
                    round(100*sqrt(power.mHSIC * (1-power
                        .mHSIC)/B), 1)
                   )
      }

    } # Case nbr.rv ==3



    # Bind those results to the previous results (from
        other sample sizes)
    # A data.frame is required to produce nice tables
        directly exportable in Latex using the 'xtable'
        package
    data <- rbind.data.frame(data, results, row.names =
       NULL)

  } # End of For loop in k (done for each sample size)

  # Name the columns
  if(nbr.rv == 2){
    colnames(data) <- c("N", "BLAW", "(sd)", "HHG", "(sd)
       ", "DCOV", "(sd)", "HSIC", "(sd)")
  }  else if(nbr.rv == 3 && v.dim == 1){
      colnames(data) <- c("N", "mBLAW", "(sd)", "mHHG", "(
          sd)", "BBL", "(sd)", "mHSIC", "(sd)", "GR", "(sd)
          ")
  }  else{
      colnames(data) <- c("N", "mBLAW", "(sd)", "mHHG", "(
          sd)", "mHSIC", "(sd)")
     }

  # Save data
  saveRDS(data, file = paste(data.direct, data.name, ".Rda
     ", sep = ""))
```

```
# Produce and save a graph of the Power vs Sample size
if(nbr.rv == 2){
  pdf(paste(graphs.direct, data.name, ".pdf", sep = ""),
     width = 10, height = 10)
  par(mar = c(5, 5, 3, 3))
  plot(data$N, data$BLAW, xlim=c(0, max(N)), ylim=c(0,
    100), type = 'l', lty = 1, lwd = 10, col="red",
    xlab="Sample size", ylab="Power", cex.lab = 2, cex.
    axis = 2)
  lines(data$N, data$HHG,  lwd = 10, lty = 2, col="blue
    ")
  lines(data$N, data$DCOV, lwd = 10, lty = 3, col="
    darkgreen")
  lines(data$N, data$HSIC, lwd = 10, lty = 4, col="
    orange")
  legend(legend.pos, NULL, ncol=1, legend=c("BLAW","HHG
    ", "DCOV", "HSIC"), col=c("red","blue", "darkgreen
    ",  "orange"), lty = c(1, 2, 3, 4), lwd = 10, cex =
     2)
}
else{
  pdf(paste(graphs.direct, data.name, ".pdf", sep = ""),
     width = 10, height = 10)
  par(mar = c(5, 5, 3, 3))
  plot(data$N, data$mBLAW, xlim=c(0, max(N)), ylim=c(0,
    100), type = 'l', lty = 1, lwd = 10, col="red",
    xlab="Sample size", ylab="Power", cex.lab = 2, cex.
    axis = 2)
  lines(data$N, data$mHHG,  lwd = 10, lty = 2, col="blue
    ")
  lines(data$N, data$mHSIC, lwd = 10, lty = 4, col="
    orange")
   if(v.dim == 1){
```

```
        lines(data$N, data$BBL, lwd = 10, lty = 3, col="
           darkgreen")
        lines(data$N, data$GR, lwd = 10, lty = 6, col="
           purple")
        legend(legend.pos, NULL, ncol=1, legend=c("mBLAW
           ","mHHG", "BBL", "mHSIC", "GR"), col=c("red","
           blue", "darkgreen",  "orange", "purple"), lty =
            c(1, 2, 3, 4, 6), lwd = 10, cex = 2)
      }
      else {
        legend(legend.pos, NULL, ncol=1, legend=c("mBLAW
           ","mHHG", "mHSIC"), col=c("red","blue",  "
           orange"), lty = c(1, 2, 4), lwd = 10, cex = 2)
      }
    }
    dev.off()


# Produce a table in the Latex syntax (and print it in
   the R console)
table <- xtable(data, caption = c(table.caption))
digits(table) <- xdigits(table)
if(nbr.rv == 3 && v.dim == 1){
  align(table) <- c("c","c|","c","c|","c","c|","c","c
     |","c","c|","c","c")
}
else if (nbr.rv == 3 && v.dim > 1){
  align(table) <- c("c","c|","c","c|","c","c|","c","c")
}
else{
  align(table) <- c("c","c|","c","c|","c","c|","c","c
     |","c","c")
}
```

```
  print(table, file = paste(data.direct, data.name, ".txt
     ", sep = ""), type="latex", include.rownames = FALSE,
      floating = TRUE, latex.environments = "center")

} # End of function
```