

Elements of Copula Modeling with R

<https://copula.r-forge.r-project.org/book/>

CRoNoS Winter Course on Copula-based modeling with R, London, 2017

Marius Hofert, Ivan Kojadinovic, Martin Mächler, Jun Yan

Overview

1 Introduction

2 Copulas

3 Classes and families

4 Estimation

5 Graphical diagnostics, tests and model selection

6 Ties, time series and regression

1 Introduction

- 1.1 A motivating example
- 1.2 The probability and quantile transformations
- 1.3 Copulas
- 1.4 Structure and philosophy
- 1.5 Additional references

1.1 A motivating example

Compare the data sets (x_{i1}, x_{i2}) , $i \in \{1, \dots, n\}$, and (y_{i1}, y_{i2}) , $i \in \{1, \dots, n\}$ in terms of the “dependence” between the two components:

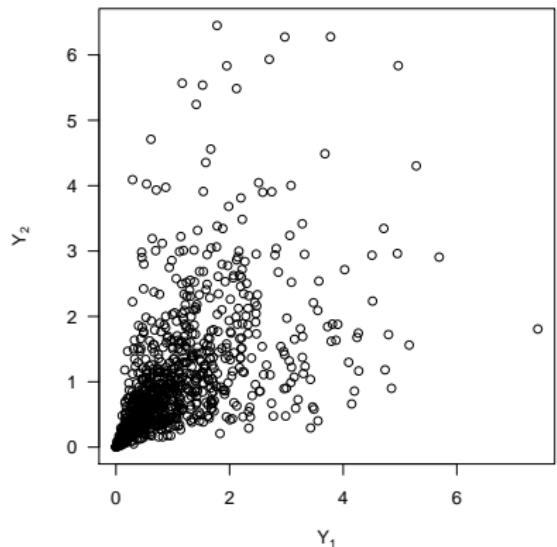
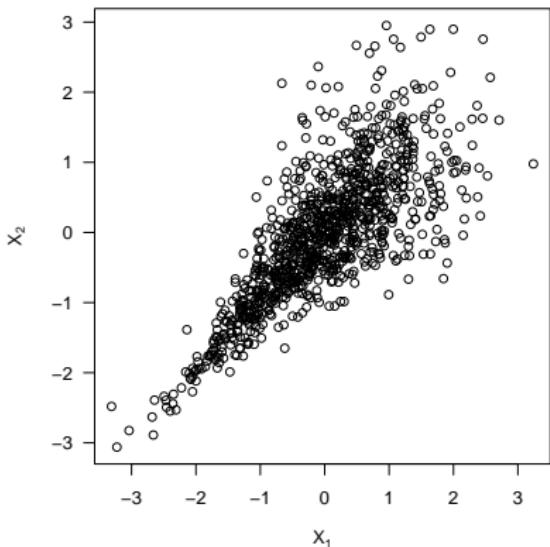


Figure 1.1 Scatter plots of $n = 1000$ *independent observations* (that is, a realization of independent copies) of (X_1, X_2) (left) and of (Y_1, Y_2) (right).

- Comparing the two data sets in terms of dependence means comparing the way X_1 and X_2 are related with the way Y_1 and Y_2 are related.
- The scatter plots do **not reveal the presence of ties in the component samples** x_{1j}, \dots, x_{nj} and y_{1j}, \dots, y_{nj} , $j \in \{1, 2\}$, of the data sets suggesting that (X_1, X_2) and (Y_1, Y_2) are ***continuous random vectors***.
- Saying that (X_1, X_2) is **continuous** means that its ***distribution function (df)*** H defined by $H(\mathbf{x}) = \mathbb{P}(X_1 \leq x_1, X_2 \leq x_2)$, $\mathbf{x} \in \mathbb{R}^2$, (where \mathbb{P} denotes the underlying probability measure), **is continuous**.
- **Equivalently**, that the univariate dfs F_1 and F_2 defined by $F_1(x) = \mathbb{P}(X_1 \leq x)$ and $F_2(x) = \mathbb{P}(X_2 \leq x)$, $x \in \mathbb{R}$, respectively, **are continuous**.
- Note that F_1 and F_2 can be recovered from H by $F_1(x) = H(x, \infty) = \lim_{x_2 \rightarrow \infty} H(x, x_2)$ and $F_2(x) = H(\infty, x) = \lim_{x_1 \rightarrow \infty} H(x_1, x)$, $x \in \mathbb{R}$, and are thus also called the ***margins of H*** or ***marginal dfs*** of (X_1, X_2) .

- Shall we estimate the linear correlation coefficient between X_1 and X_2 , and compare it with the one between Y_1 and Y_2 ?
- A standard computation gives approximately 0.77 (respectively, 0.56) for X_1 and X_2 (respectively, Y_1 and Y_2).
- This seems to indicate that the dependence between X_1 and X_2 is stronger than the dependence between Y_1 and Y_2 . This conclusion does not appear unrealistic in view of Figure 1.1.
- However, linear correlation only captures the linear dependence in the two bivariate samples. Are the relationships between the variables in the left and right panels of Figure 1.1 only of a linear nature? Since the dispersion increases as one of the two variables increases, clearly not.
- What was done thus far may therefore look quite unsatisfactory. After some further thinking, one might decide to assess what the marginal distributions of (X_1, X_2) and (Y_1, Y_2) (that is, the distributions of X_1, X_2, Y_1, Y_2 separately) look like.

- X_1 and X_2 seem to follow a standard normal distribution $N(0, 1)$, while Y_1 and Y_2 seem to follow a standard exponential distribution $\text{Exp}(1)$:

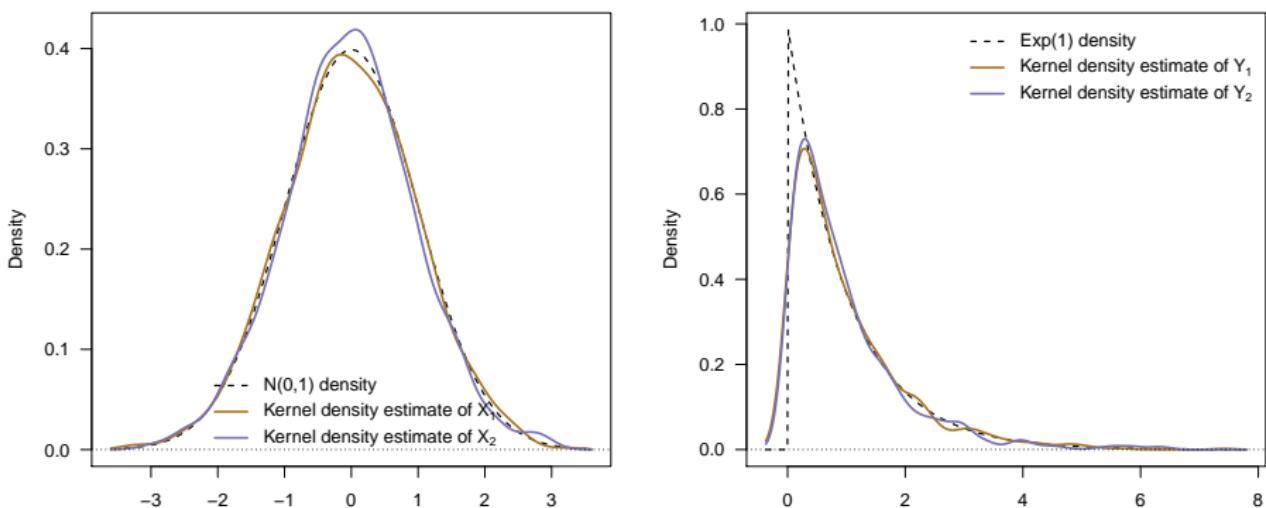


Figure 1.2 Kernel density estimates of the densities of X_1, X_2 (left) and Y_1, Y_2 (right). The dashed curves represent the $N(0, 1)$ (left) and $\text{Exp}(1)$ (right) densities.

- The fact that the two data sets in Figure 1.1 differ at least marginally clearly affects one's perception of the possible difference in dependence.

- If we could transform the two data sets so that they become similar in terms of the underlying marginal dfs, their comparison in terms of dependence would be made on much fairer grounds.

1.2 The probability and quantile transformations

One transformation we could use is that to a standard uniform distribution, known as the *probability (integral) transformation*.

Lemma 1.2.1 (Probability transformation)

Let F be a continuous df and $X \sim F$. Then $F(X)$ is a standard uniform random variable, that is, $F(X) \sim U(0, 1)$. □

- The probability transformation transforms a random variable with continuous df F to a standard uniform random variable.
- The continuity of F is crucial as otherwise the range of F would not contain $(0, 1)$.

- Provided that the marginal dfs of the underlying random vectors (X_1, X_2) and (Y_1, Y_2) were known, the two data sets represented in Figure 1.1 could be transformed into data sets consisting of observations of random vectors with standard uniform univariate marginal dfs, thereby making their comparison in terms of dependence fairer.
- Specifically, let F_1 , F_2 , G_1 and G_2 denote the dfs of X_1 , X_2 , Y_1 and Y_2 , respectively, and let us assume for the moment that $F_1 = F_2$ is the $N(0, 1)$ df and $G_1 = G_2$ is the $\text{Exp}(1)$ df.
- The corresponding realizations of $(F_1(X_1), F_2(X_2))$ and $(G_1(Y_1), G_2(Y_2))$ are simply obtained by applying F_1 to the first component sample of the first data set (so to the realizations of X_1) and F_2 to the second component sample of the first data set (so to the realizations of X_2); similarly for the second data set.

- Scatter plots of the transformed data sets are displayed in Figure 1.3.

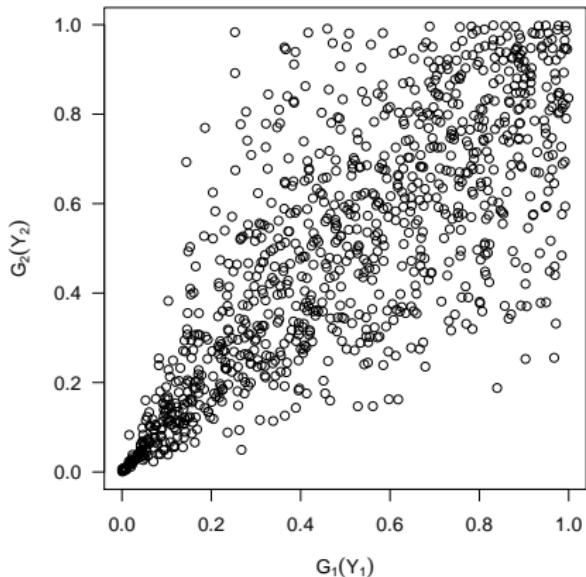
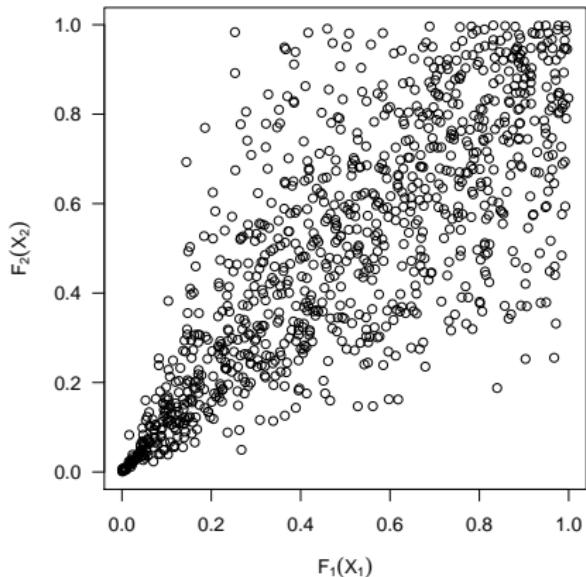


Figure 1.3 Scatter plots of $n = 1000$ independent observations of the bivariate random vectors $(F_1(X_1), F_2(X_2))$ (left) and $(G_1(Y_1), G_2(Y_2))$ (right).

- The transformed data sets look similar (in fact, it follows from how the two data sets were constructed that they are actually identical).

- The distributions of $(F_1(X_1), F_2(X_2))$ and $(G_1(Y_1), G_2(Y_2))$ seem to be identical.
- If one accepts that the (for the moment vague) notion of dependence between the components of a continuous random vector should not be affected by its marginal distributions (a special case of an invariance principle; see Section 2.4), the conclusion is that the two data sets in Figure 1.1 are indistinguishable in terms of dependence and only differ in terms of the underlying marginal dfs.
- An alternative solution, for instance, would have been to transform the second data set so that it consists of observations of a bivariate random vector with standard normal margins.
- To this end, one needs the “converse” transformation of the probability transformation. It is called the *quantile transformation* and is given in Lemma 1.2.2 below.

- This transformation lies at the heart of the *inversion method* for pseudo-random number generation from non-uniform distributions; see Devroye (1986).
- To any df F is associated a *quantile function* F^\leftarrow defined by

$$F^\leftarrow(y) = \inf\{x \in \mathbb{R} : F(x) \geq y\}, \quad y \in [0, 1], \quad (1.1)$$

where \inf stands for the infimum with the convention that $\inf \emptyset = \infty$.

- This definition extends to increasing functions, providing a notion of an inverse called *generalized inverse*. For continuous and strictly increasing dfs F , F^\leftarrow equals the ordinary inverse F^{-1} . One can often work with F^\leftarrow as if it was F^{-1} ; see Embrechts and Hofert (2013).

Lemma 1.2.2 (Quantile transformation)

Let $U \sim U(0, 1)$ and let F be any df. Then, the random variable $F^\leftarrow(U)$ has df F , that is, $F^\leftarrow(U) \sim F$. □

- The quantile transformation transforms $U(0, 1)$ -variates into variates from a distribution with df F . In contrast to F in Lemma 1.2.1, F in Lemma 1.2.2 does not need to be continuous.
- To apply this result to transform (Y_1, Y_2) to a bivariate random vector with standard normal margins, one would first form $(G_1(Y_1), G_2(Y_2))$ (thus using Lemma 1.2.1) and then $(F_1^\leftarrow(G_1(Y_1)), F_2^\leftarrow(G_2(Y_2)))'$ (thus using Lemma 1.2.2).
- To obtain the corresponding realizations, one simply needs to apply $F_1^\leftarrow(G_1(\cdot))$ and $F_2^\leftarrow(G_2(\cdot))$ to the first and second component samples, respectively, of the second data set.
- The resulting scatter plot is shown on the right-hand side of Figure 1.4.

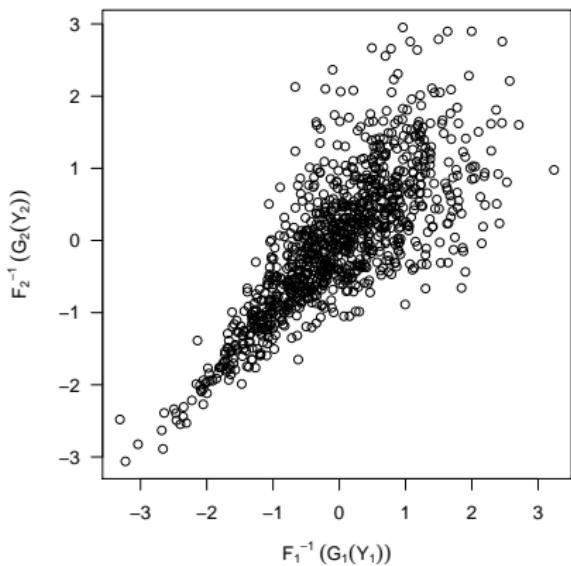
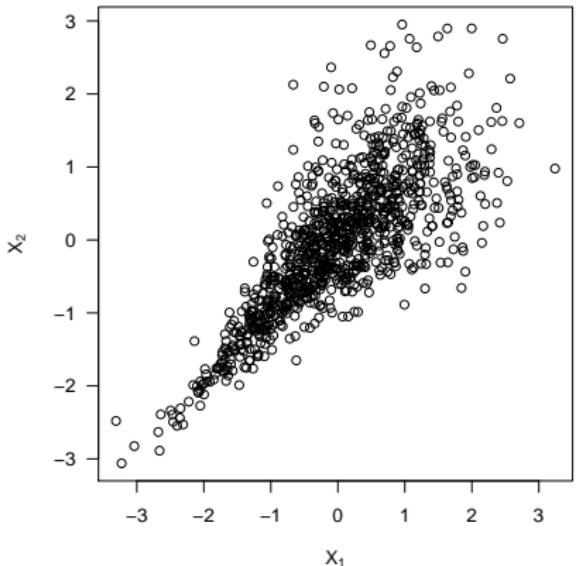


Figure 1.4 Scatter plots of $n = 1000$ independent observations of the bivariate random vectors (X_1, X_2) (left) and $(F_1^{-1}(G_1(Y_1)), F_1^{-1}(G_2(Y_2)))$ (right).

- As before, we see that the samples (which consist of independent observations of (X_1, X_2) and $(F_1^{-1}(G_1(Y_1)), F_1^{-1}(G_2(Y_2)))$, respectively) are equal. We can thus (again) conclude that the two data sets only differ in terms of their marginal dfs but are equal in terms of “dependence”.

1.3 Copulas

- The notion of dependence will coincide with that of a *copula*, that is, a multivariate df with standard uniform univariate margins.
- The copula of (X_1, X_2) and the copula of (Y_1, Y_2) are simply the dfs of $(F_1(X_1), F_2(X_2))$ and $(G_1(Y_1), G_2(Y_2))$, respectively; this follows formally from Lemma 1.2.1 and the aforementioned *invariance principle*.
- The statement that " (X_1, X_2) and (Y_1, Y_2) have the same dependence" can then be rephrased as " (X_1, X_2) and (Y_1, Y_2) have the same copula".
- The marginal dfs were assumed to be known. In practice, sample analogues of the probability and quantile transformations can be applied.
- The requirement that the margins be standard uniform is somewhat arbitrary. From the previous derivations, we see that the same conclusion for the two data sets is reached when the "standardization" is made to standard normal margins instead.

- Due to the probability and quantile transformations, $\text{U}(0, 1)$ margins turn out to be a natural and convenient choice.
- Copulas are applied in a wide variety of areas such as quantitative risk management, finance, econometric modeling or environmental modeling, to name a very few; see, for example McNeil et al. (2015), Patton (2013), Salvadori et al. (2007) and Genest et al. (2009b).
- The reason for this lies in **Sklar's Theorem** (which will be precisely stated in Section 2.3). Let (X_1, \dots, X_d) be a d -dimensional random vector and let $H(\mathbf{x}) = \mathbb{P}(X_1 \leq x_1, \dots, X_d \leq x_d)$, $\mathbf{x} \in \mathbb{R}^d$, be its df.
- The first part of Sklar's Theorem asserts that every d -dimensional df H can be expressed as

$$H(\mathbf{x}) = C(F_1(x_1), \dots, F_d(x_d)), \quad \mathbf{x} \in \mathbb{R}^d, \quad (1.2)$$

in terms of a d -dimensional copula C and the univariate marginal dfs F_1, \dots, F_d obtained from H by $F_j(x_j) = H(\infty, \dots, \infty, x_j, \infty, \dots, \infty)$, $x_j \in \mathbb{R}$.

- The copula C is thus the function which connects or couples the marginal dfs F_1, \dots, F_d to the multivariate df H , hence the name “copula”.
- For estimation of H from data, this offers a great deal of flexibility as it allows one to model F_1, \dots, F_d separately from the dependence/copula C (often of interest from a statistical and numerical point of view).
- The second part of Sklar's Theorem provides a converse. Given any copula C and univariate dfs F_1, \dots, F_d , a multivariate df H can be composed via (1.2) which has univariate margins F_1, \dots, F_d and “dependence structure” C .
- This is of interest in applications such as stress testing in finance, insurance and quantitative risk management, and in model building in general.

1.4 Structure and philosophy

- Chapter 2 offers a basic introduction to copulas and their properties.
- Chapter 3 introduces the main copula classes and the corresponding sampling procedures, along with some copula transformations.
- The estimation of copulas is addressed in Chapter 4 from a parametric, semi-parametric and non-parametric perspective.
- Chapter 5 presents graphical diagnostics, statistical tests and model selection.
- Chapter 6 is concerned with more advanced topics such as the handling of ties, time series and covariates (in a regression-like setting).

Overall, the theoretical concepts introduced are illustrated by numerous R examples using functions predominantly from the R package `copula`. More information is available under <https://copula.r-forge.r-project.org/book/>.

1.5 Additional references

- Early monographs on copulas are Joe (1997) (with focus on novel probabilistic notions around copulas) and Nelsen (2006) (a well-known, readable introduction).
- An interesting historical perspective and introduction can be found in Durante and Sempi (2010). A more advanced probabilistic treatment of copulas is the recent Durante and Sempi (2015).
- An overview paper of the theory and practice with financial applications in mind is given by Embrechts (2009). An important article well-known for warning probabilists, statisticians and financial data analysts alike about fallacies related to the use of linear correlation to measure dependence is Embrechts et al. (2002). For a more in-depth introductory treatment of copulas also stressing the latter topics in the context of quantitative risk management, see McNeil et al. (2005, Chapter 5) or McNeil et al. (2015, Chapter 7).

- Monographs on copulas mainly targeted towards the realm of finance are Cherubini et al. (2004), Cherubini et al. (2011) and Mai and Scherer (2014); see also Rémillard (2013, Chapter 8).
- An easy-to-digest and statistically oriented introduction to copulas is given by Genest and Favre (2007); see also Genest et al. (2009a) for additional material on goodness-of-fit testing. A recent statistically-oriented monograph is Joe (2014).
- A currently active area of research not covered in this course are vine copulas; see <http://vine-copula.org> for more details including publications and research projects.

2 Copulas

- 2.1 Definition and characterization
- 2.2 The Fréchet–Höffding bounds
- 2.3 Sklar's Theorem
- 2.4 The invariance principle
- 2.5 Survival copulas and copula symmetries
- 2.6 Measures of association

2.1 Definition and characterization

- Recall that the **df H** of a d -dimensional random vector $\mathbf{X} = (X_1, \dots, X_d)$ is the function defined by

$$H(\mathbf{x}) = \mathbb{P}(\mathbf{X} \leq \mathbf{x}) = \mathbb{P}(X_1 \leq x_1, \dots, X_d \leq x_d),$$

for all $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$.

- The **df F_j of X_j** , $j \in \{1, \dots, d\}$, can be recovered from H by

$$F_j(x_j) = H(\infty, \dots, \infty, x_j, \infty, \dots, \infty), \quad x_j \in \mathbb{R};$$

this is why F_1, \dots, F_d are also called the *univariate margins of H* or the *marginal dfs of \mathbf{X}* .

Definition 2.1.1 (Copula)

A **copula** is a multivariate **df** with standard uniform univariate margins, that is, **$U(0, 1)$ margins**. □

- As mentioned in Chapter 1, the choice of $U(0, 1)$ margins is not necessary but in light of Lemmas 1.2.1 and 1.2.2 a rather sensible one.
- One of the simplest copulas is the *independence copula*

$$\Pi(\mathbf{u}) = \prod_{j=1}^d u_j, \quad \mathbf{u} \in [0, 1]^d, \tag{2.1}$$

which is the df of a random vector $\mathbf{U} = (U_1, \dots, U_d)$ with $U_1, \dots, U_d \stackrel{\text{ind.}}{\sim} U(0, 1)$; this follows from $\mathbb{P}(\mathbf{U} \leq \mathbf{u}) = \mathbb{P}(U_1 \leq u_1, \dots, U_d \leq u_d) = \prod_{j=1}^d \mathbb{P}(U_j \leq u_j) = \prod_{j=1}^d u_j = \Pi(\mathbf{u}), \mathbf{u} \in [0, 1]^d$.

Example 2.1.2 (Independence copula)

In the R package `copula`, the copula Π is represented by an R object created using the function `indepCopula()`:

```
1 > library(copula)
2 > d <- 2
3 > ic <- indepCopula(dim = d)
```

To evaluate the copula Π , the function `pCopula()` can be used:

```
1 > set.seed(2008)
2 > u <- runif(d) # a random point in the unit hypercube
3 > (Pi <- pCopula(u, copula = ic)) # the value of the independence copula at u
[1] 0.3668219
```

Numerical equality (between the computation carried out with `pCopula()` and directly with (2.1)) can be checked as follows:

```
1 > stopifnot(all.equal(Pi, prod(u))) # check numerical equality of the samples
```

When $d = 2$, surface (or perspective, or wireframe) plots and contour (or level) plots of Π can be produced with `wireframe2()` and with `contourplot2()`:

```
1 > wireframe2 (ic, FUN = pCopula, # surface plot of the independence copula
2                 col.4 = adjustcolor("black", alpha.f = 0.25))
3 > contourplot2(ic, FUN = pCopula) # contour plot of the independence copula
```

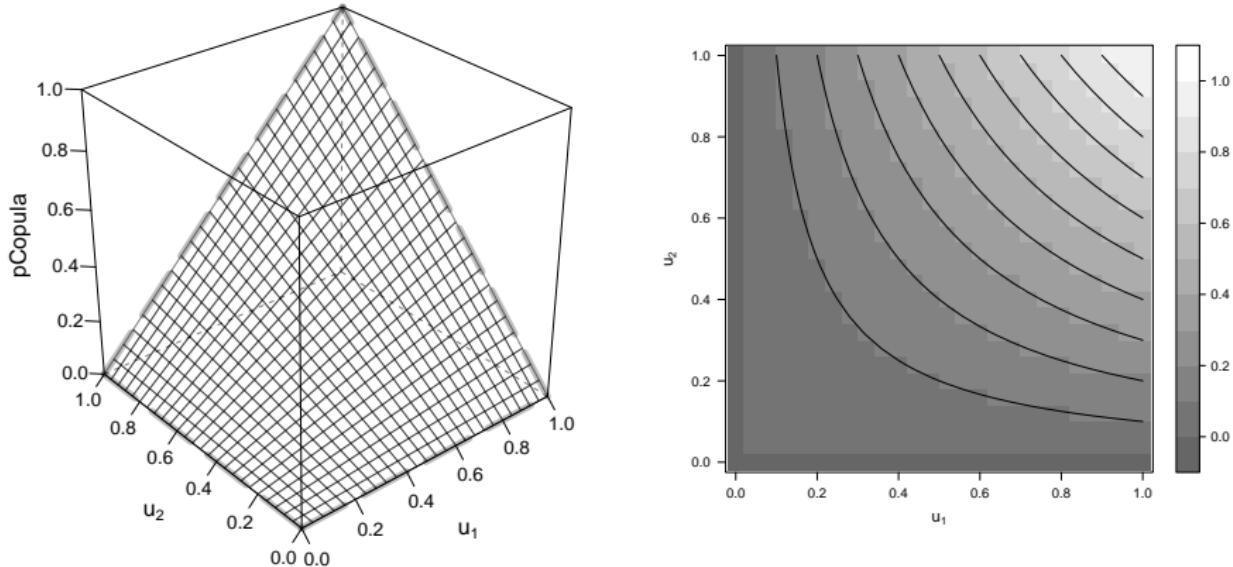


Figure 2.1 Wireframe plot (left) and contour plot (right) of the independence copula. □

- As we can see ($d = 2$) for Π , any d -dimensional copula C , $d \geq 2$,
 - 1) ... is *grounded* ($C(\mathbf{u}) = 0$ if $u_j = 0$ for at least one $j \in \{1, \dots, d\}$);
 - 2) ... has *standard uniform univariate margins* (for any $j \in \{1, \dots, d\}$, $C(1, \dots, 1, u_j, 1, \dots, 1) = u_j$ for all $u_j \in [0, 1]$).

- These two properties alone are not sufficient for C to be a df and thus a copula; the function W in (2.6) for $d \geq 3$ provides a counter-example.
- One needs one additional property. To introduce it, we need the following additional definitions. For any $\mathbf{a}, \mathbf{b} \in [0, 1]^d$, $\mathbf{a} \leq \mathbf{b}$, let $(\mathbf{a}, \mathbf{b}]$ denote the hyperrectangle $\{\mathbf{u} \in [0, 1]^d : \mathbf{a} < \mathbf{u} \leq \mathbf{b}\}$. Then, for any $(\mathbf{a}, \mathbf{b}]$ in $[0, 1]^d$, define its *C-volume* as

$$\Delta_{(\mathbf{a}, \mathbf{b}]} C = \sum_{\mathbf{i} \in \{0,1\}^d} (-1)^{\sum_{j=1}^d i_j} C(a_1^{i_1} b_1^{1-i_1}, \dots, a_d^{i_d} b_d^{1-i_d}), \quad (2.2)$$

where the summation is over all $\mathbf{i} = (i_1, \dots, i_d)$ for $i_1, \dots, i_d \in \{0, 1\}$.

- If the C -volumes of all hyperrectangles $(\mathbf{a}, \mathbf{b}]$ in $[0, 1]^d$ are non-negative, that is,

$$\Delta_{(\mathbf{a}, \mathbf{b}]} C \geq 0 \quad \text{for all } \mathbf{a}, \mathbf{b} \in [0, 1]^d, \mathbf{a} \leq \mathbf{b},$$

C is called *d-increasing*.

- C -volumes have a natural interpretation. For $\mathbf{U} \sim C$, it can be verified that

$$\Delta_{(\mathbf{a}, \mathbf{b})} C = \mathbb{P}(\mathbf{U} \in (\mathbf{a}, \mathbf{b})) \quad \text{for all hyperrectangles } (\mathbf{a}, \mathbf{b}) \text{ in } [0, 1]^d. \quad (2.3)$$

- When $d = 2$, (2.2) becomes

$$\Delta_{(\mathbf{a}, \mathbf{b})} C = C(b_1, b_2) - C(b_1, a_2) - C(a_1, b_2) + C(a_1, a_2).$$

- Let $C = \Pi$ and let us verify (2.3).

- ▶ On the one hand, we immediately have that

$$\begin{aligned} \Delta_{(\mathbf{a}, \mathbf{b})} \Pi &= b_1 b_2 - b_1 a_2 - a_1 b_2 + a_1 a_2 \\ &= b_1(b_2 - a_2) - a_1(b_2 - a_2) = (b_1 - a_1)(b_2 - a_2). \end{aligned}$$

- ▶ On the other hand, we have that

$$\begin{aligned} \mathbb{P}(\mathbf{U} \in (\mathbf{a}, \mathbf{b})) &= \mathbb{P}(a_1 < U_1 \leq b_1) \mathbb{P}(a_2 < U_2 \leq b_2) \\ &= (b_1 - a_1)(b_2 - a_2). \end{aligned}$$

Example 2.1.3 (*C*-volumes)

In the R package `copula`, *C*-volumes can be computed using the function `prob()`. The following code illustrates the computation of $\Delta_{(a,b)}\Pi$ for $a = (1/4, 1/2)$ and $b = (1/3, 1)$:

```
1 > a <- c(1/4, 1/2) # lower left end point
2 > b <- c(1/3, 1) # upper right end point
3 > stopifnot(0 <= a, a <= 1, 0 <= b, b <= 1, a <= b) # check
4 > p <- (b[1] - a[1]) * (b[2] - a[2]) # manual computation
5 > stopifnot(all.equal(prob(ic, l = a, u = b), p)) # check
```

We can also obtain approximations of *C*-volumes by simulation using the function `rCopula()` which generates independent observations from a given copula. We shall illustrate this in the case of the previous Π -volume below. But first, let us generate and plot a sample from Π (see Figure 2.2):

```
1 > n <- 1000 # sample size
2 > set.seed(271) # set a seed (for reproducibility)
3 > U <- rCopula(n, copula = ic) # generate a sample of the independence copula
4 > plot(U, xlab = quote(U[1]), ylab = quote(U[2]))
```

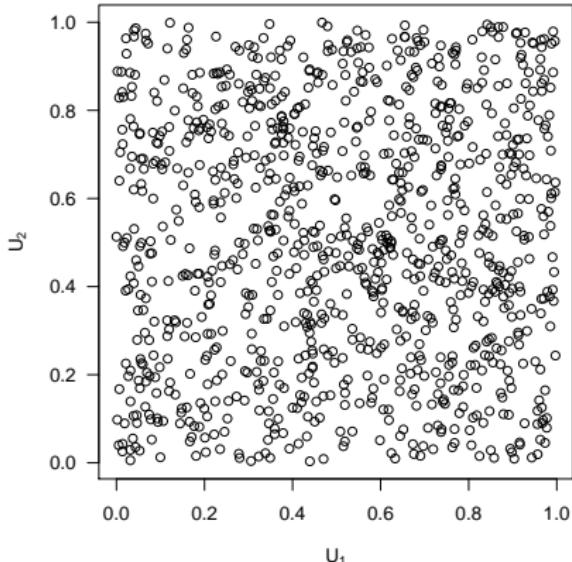


Figure 2.2 $n = 1000$ independent observations of $U \sim \Pi$.

Note that `rCopula(n, ic)` is merely a wrapper for `matrix(runif(n * d), nrow = n)`:

```
1 > set.seed(271)
2 > stopifnot(all.equal(U, matrix(runif(n * d), nrow = n)))
```

The Π -volume of $(1/4, 1/3] \times (1/2, 1]$ can then be approximated by the proportion of realizations of $\mathbf{U} \sim \Pi$ falling in this hyperrectangle:

```
1 > set.seed(314)
2 > U <- rCopula(1e6, copula = ic) # large sample size for good approximation
3 > ## Approximate the Pi-volume by the aforementioned proportion
4 > p.sim <- mean(a[1] < U[,1] & U[,1] <= b[1] & a[2] < U[,2] & U[,2] <= b[2])
5 > stopifnot(all.equal(p.sim, p, tol = 1e-2)) # note: may depend on seed
```



- A copula C is called *absolutely continuous* if it admits a *density*, that is, if

$$c(\mathbf{u}) = \frac{\partial^d}{\partial u_d \dots \partial u_1} C(u_1, \dots, u_d), \quad \mathbf{u} \in (0, 1)^d, \quad (2.4)$$

exists.

- In the R package `copula`, copula densities can be evaluated with the function `dCopula()`.
- If, for a function $C : [0, 1]^d \rightarrow [0, 1]$, the right-hand side of (2.4) is non-negative for all $\mathbf{u} \in (0, 1)^d$, then the function is *d-increasing*.

- As an example, the independence copula Π is absolutely continuous with constant density $c(\mathbf{u}) = 1$, $\mathbf{u} \in (0, 1)^d$; note the similarity to the univariate case ($U(0, 1)$ has the constant density 1 on $(0, 1)$).

In the same way that parametric univariate distributions are crucial for univariate statistics, parametric copula families play a key role in applications of copulas. We now introduce some parametric copula families.

Example 2.1.4 (Frank copula)

As a first example, consider the *Frank family*; see Genest (1987). Its members, parametrized by $\theta \in \mathbb{R} \setminus \{0\}$ are, in the bivariate case, the copulas defined by

$$C_\theta^F(\mathbf{u}) = -\frac{1}{\theta} \log \left(1 + \frac{(\exp(-\theta u_1) - 1)(\exp(-\theta u_2) - 1)}{(\exp(-\theta) - 1)} \right), \quad \mathbf{u} \in [0, 1]^2, \tag{2.5}$$

with the convention that $C_0^F = \Pi$, as a consequence of the fact that the function in (2.5) converges to Π as $\theta \rightarrow 0$.

- Two immediate consequences from (2.5) are **groundedness** (for example, if $u_1 = 0$, then $C_\theta^F(\mathbf{u}) = -\log(1 + 0)/\theta = 0$) and **standard uniform univariate margins** (for example, if $u_2 = 1$, then $C_\theta^F(\mathbf{u}) = -\log(1 + (\exp(-\theta u_1) - 1))/\theta = u_1$).
- **2-increasingness can also be shown** (for example, by deriving the density c_θ^F and showing that it is indeed non-negative).
- The higher-dimensional version of C_θ^F will be defined in Chapter 3.
- The parameter θ controls the dependence between the components of $\mathbf{U} \sim C_\theta^F$ as will be illustrated later in this example.

In the R package `copula`, the Frank copula C_θ^F is represented by an R object created using the function `frankCopula()`:

```

1 > d <- 2 # dimension
2 > theta <- -9 # copula parameter
3 > fc <- frankCopula(theta, dim = d) # define a Frank copula

```

The created object `fc` thus represents a Frank copula with parameter $\theta = -9$. Its $\text{df } C_\theta^F$ and corresponding density c_θ^F can be computed with the functions `pCopula()` and `dCopula()`, respectively:

```
1 > set.seed(2010)
2 > n <- 5 # number of evaluation points
3 > u <- matrix(runif(n * d), nrow = n) # n random points in [0,1]^d
4 > pCopula(u, copula = fc) # copula values at u

[1] 0.2515825004 0.1438027127 0.0005267892 0.0053807993 0.0075792044

1 > dCopula(u, copula = fc) # density values at u

[1] 0.9012539 1.8161995 0.1400732 0.4590708 2.4612347
```

Wireframe and contour plots of the functions C_θ^F and the corresponding density c_θ^F (for $d = 2$ and $\theta = -9$; see Figure 2.3) can be easily produced:

```
1 > wireframe2(fc, FUN = pCopula, # wireframe plot (copula)
   draw.4.pCoplines = FALSE)
2 > wireframe2(fc, FUN = dCopula, delta = 0.001) # wireframe plot (density)
3 > contourplot2(fc, FUN = pCopula) # contour plot (copula)
4 > contourplot2(fc, FUN = dCopula, n.grid = 72, # contour plot (density)
   lwd = 1/2)
```

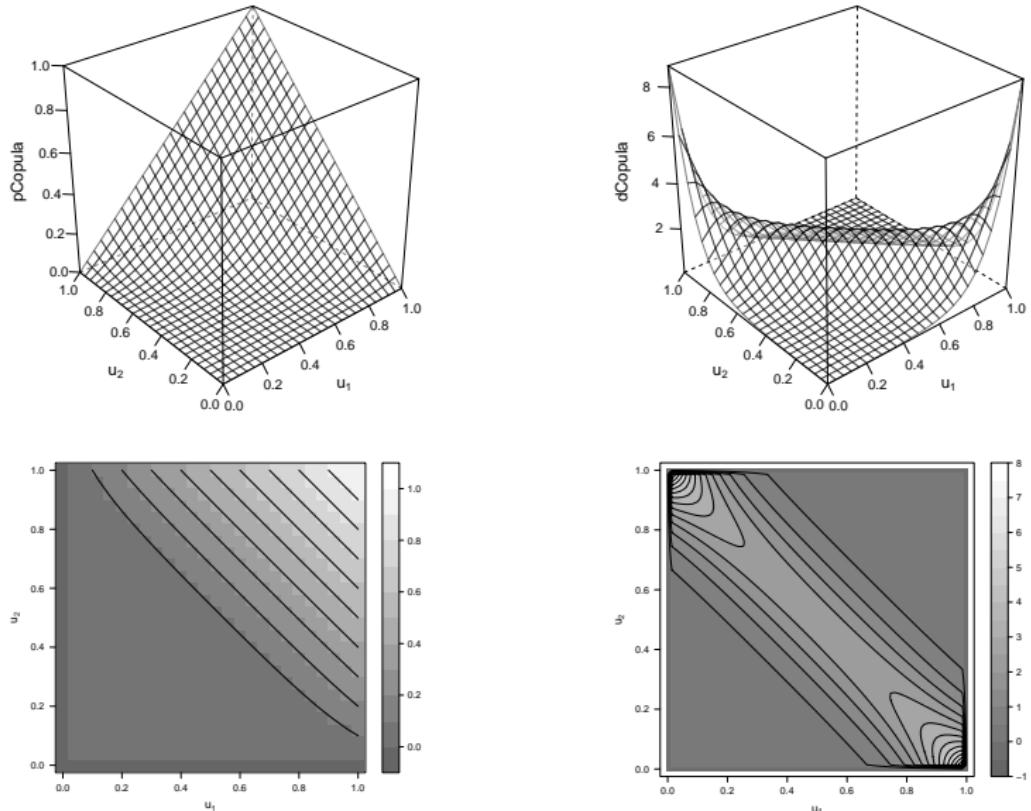


Figure 2.3 Wireframe (top) and contour plots (bottom) of C_θ^F (left) and of the corresponding density c_θ (right) for $d = 2$ and $\theta = -9$.

`rCopula()` can be used to generate $n = 1000$ independent observations from C_θ^F for $\theta \in \{-9, 0, 9\}$ ($\theta = 0$ corresponds to independence):

```
1 > set.seed(1946)
2 > n <- 1000
3 > U <- rCopula(n, copula = fc)
4 > U0 <- rCopula(n, copula = setTheta(fc, value = 0)) # setTheta() changes the parameter
5 > U9 <- rCopula(n, copula = setTheta(fc, value = 9))
6 > plot(U, xlab = quote(U[1]), ylab = quote(U[2]))
7 > plot(U0, xlab = quote(U[1]), ylab = quote(U[2]))
8 > plot(U9, xlab = quote(U[1]), ylab = quote(U[2]))
```

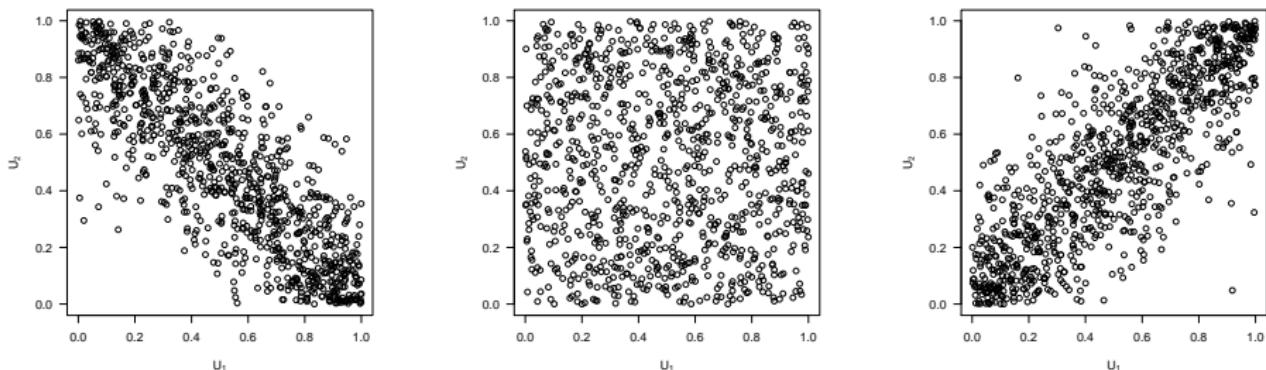


Figure 2.4 $n = 1000$ independent observations of $(U_1, U_2) \sim C_\theta^F$ for $\theta = -9$ (left), $\theta = 0$ (middle) and $\theta = 9$ (right).

- Increasing θ from 0 to 9, the components U_1 and U_2 of $\mathbf{U} \sim C_\theta^F$ become *positively dependent* in the sense that small (respectively, large) values of U_1 tend to be associated with small (respectively, large) values of U_2 .
- For $\theta = -9$, U_1 and U_2 are *negatively dependent* in the sense that small (respectively, large) values of U_1 tend to be associated with large (respectively, small) values of U_2 . \square

Example 2.1.5 (Clayton copula)

- For $d = 2$, the members of the *Clayton* family, parametrized by $\theta \in [-1, \infty) \setminus \{0\}$, are defined by

$$C_\theta^C(\mathbf{u}) = \max\{u_1^{-\theta} + u_2^{-\theta} - 1, 0\}^{-1/\theta}, \quad \mathbf{u} \in [0, 1]^2.$$

- For $d \geq 3$, we only consider $\theta \in (0, \infty)$ and we have

$$C_\theta^C(\mathbf{u}) = \left(1 - d + \sum_{j=1}^d u_j^{-\theta}\right)^{-1/\theta}, \quad \mathbf{u} \in [0, 1]^d.$$

- Both in the bivariate and in the multivariate case, we adopt the convention that $C_0^C = \Pi$ (the latter follows naturally from the fact that C_θ^C converges to Π as $\theta \rightarrow 0$).
- For $d \geq 2$ and $\theta \in (0, \infty)$, the density is

$$c_\theta^C(\mathbf{u}) = \left(\prod_{k=0}^{d-1} (\theta k + 1) \right) \left(\prod_{j=1}^d u_j \right)^{-(1+\theta)} \left(1 - d + \sum_{j=1}^d u_j^{-\theta} \right)^{-(d+1/\theta)}.$$

R objects allowing to manipulate Clayton copulas can be created with `claytonCopula()`:

```
1 > d <- 3
2 > cc <- claytonCopula(4, dim = d) # theta = 4
```

As previously, the object can be used to evaluate C_θ^C and its corresponding density c_θ^C using the functions `pCopula()` and `dCopula()`, respectively:

```
1 > set.seed(2013)
2 > n <- 5
3 > u <- matrix(runif(n * d), nrow = n) # random points in the unit hypercube
```

```
4 > pCopula(u, copula = cc) # copula values at u  
  
[1] 0.004559638 0.247045204 0.634084005 0.702682364 0.250533621
```

```
1 > dCopula(u, copula = cc) # density values at u  
  
[1] 1.947682e-15 1.251657e-03 5.108704e+00 5.341037e+00 1.931097e-02
```

Independent observations of $U \sim C_\theta^C$ (that is, a random sample from C_θ^C) can be generated using `rCopula()`:

```
1 > set.seed(271)  
2 > U <- rCopula(1000, copula = cc)  
3 > splom2(U, cex = 0.3, col.mat = "black")
```

The resulting `scatter-plot matrix` (based on the function `splom2()` from the R package `copula`) is shown in Figure 2.5.

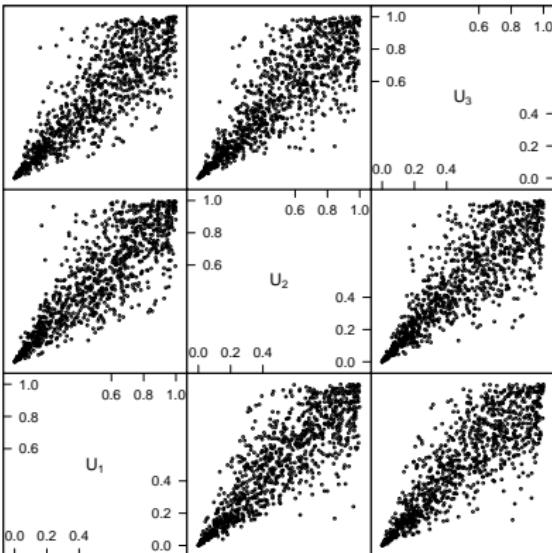


Figure 2.5 Scatter-plot matrix of $n = 1000$ independent observations from a trivariate Clayton copula C_θ^C for $\theta = 4$.

- The larger $\theta \geq 0$, the stronger the (positive) dependence between the components of $\mathbf{U} \sim C_\theta^C$.
- It can also be verified, that values of θ in $[-1, 0)$ lead to negative dependence between the components of $(U_1, U_2) \sim C_\theta^C$. □

Example 2.1.6 (Gumbel–Hougaard copula)

The members of the *Gumbel–Hougaard* family, parametrized by $\theta \in [1, \infty)$, are defined by

$$C_{\theta}^{\text{GH}}(\mathbf{u}) = \exp \left(- \left(\sum_{i=1}^d (-\log u_i)^{\theta} \right)^{1/\theta} \right), \quad \mathbf{u} \in [0, 1]^d.$$

Gumbel–Hougaard copula objects can be created via `gumbelCopula()`:

```
1 > gc <- gumbelCopula(3) # theta = 3 (note the default dim = 2)
```

Evaluation of C_{θ}^{GH} , its density c_{θ}^{GH} or random number generation from C_{θ}^{GH} can be carried out as previously with `pCopula()`, `dCopula()` and `rCopula()`. For instance:

```
1 > set.seed(1993)
2 > U <- rCopula(1000, copula = gc)
3 > plot(U, xlab = quote(U[1]), ylab = quote(U[2]))
4 > wireframe2(gc, dCopula, delta = 0.025) # wireframe plot (density)
```

The resulting plots are shown in Figure 2.6:

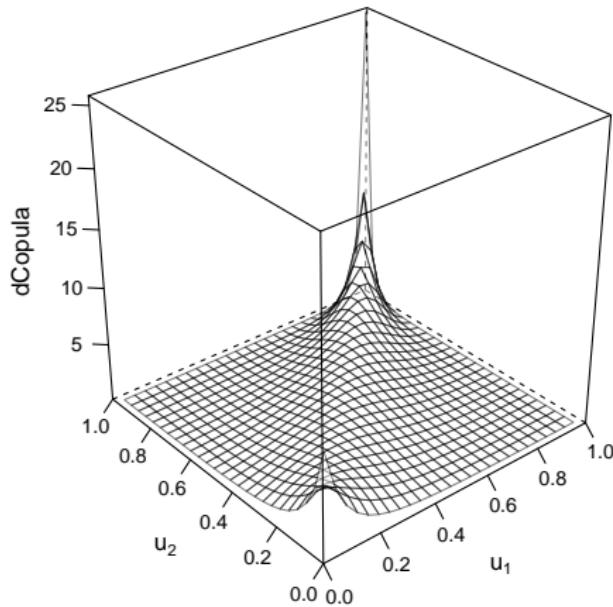
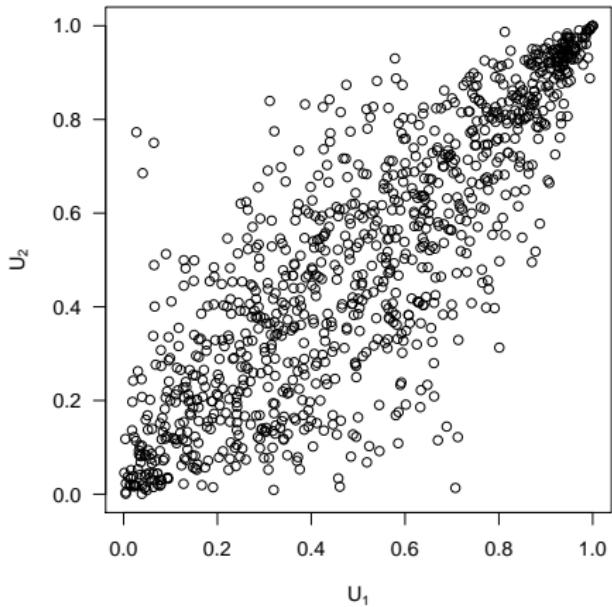


Figure 2.6 Scatter plot of $n = 1000$ independent observations from C_{θ}^{GH} for $\theta = 3$ (left) and wireframe plot of the corresponding density c_{θ}^{GH} (right).



2.2 The Fréchet–Höffding bounds

- The following theorem is one of the cornerstones of copula theory.
- For $\mathbf{u} \in [0, 1]^d$, let

$$W(\mathbf{u}) = \max \left\{ \sum_{j=1}^d u_j - d + 1, 0 \right\} \quad \text{and} \quad M(\mathbf{u}) = \min_{1 \leq j \leq d} \{u_j\}; \quad (2.6)$$

these functions are known as *lower Fréchet–Höffding bound* and *upper Fréchet–Höffding bound*, respectively.

- W is a copula only if $d = 2$ whereas M is a copula for all $d \geq 2$.

Theorem 2.2.1 (Fréchet–Höffding bounds)

For any d -dimensional copula C ,

$$W(\mathbf{u}) \leq C(\mathbf{u}) \leq M(\mathbf{u}), \quad \mathbf{u} \in [0, 1]^d. \quad \square$$

- Theorem 2.2.1 is attributed to Höffding (1940) and Fréchet (1951), already partly appearing in Fréchet (1935).

- Let $U \sim U(0, 1)$. It is easy to verify that

$$(U, 1 - U) \sim W \quad \text{and} \quad (U, \dots, U) \sim M. \quad (2.7)$$

- Due to these *stochastic representations*, that is, representations in the distributional sense by simple random variables as building blocks, W (in dimension two only) is often called the *countermonotone copula* and M is often referred to as the *comonotone copula*.
- (U, \dots, U) (modeled by M) reflects *perfect positive dependence* (in the sense that if one component increases, all the other components increase as well *almost surely*, that is, with probability 1) whereas $(U, 1 - U)$ (modeled by W) reflects *perfect negative dependence* (in the sense that if one component increases, the other component almost surely decreases).
- The notion of perfect negative dependence cannot, however, be extended to the case $d \geq 3$.

Example 2.2.2 (Fréchet–Höffding bounds)

Due to the stochastic representations (2.7), it is easy to generate a random sample from M (in any dimension) and W (in dimension two):

```
1 > set.seed(1980)
2 > U <- runif(100)
3 > plot(cbind(U, 1-U), xlab = quote(U[1]), ylab = quote(U[2]))
4 > plot(cbind(U, U),   xlab = quote(U[1]), ylab = quote(U[2]))
```

The resulting plots are shown in Figure 2.7. W and M are among the rare copulas which can be identified from their wireframe and contour plots.

They can be produced as follows:

```
1 > u <- seq(0, 1, length.out = 40) # subdivision points in each dimension
2 > u12 <- expand.grid("u[1]" = u, "u[2]" = u) # build a grid
3 > W <- pmax(u12[,1] + u12[,2] - 1, 0) # values of W on grid
4 > M <- pmin(u12[,1], u12[,2]) # values of M on grid
5 > val.W <- cbind(u12, "W(u[1],u[2])" = W) # append grid
6 > val.M <- cbind(u12, "M(u[1],u[2])" = M) # append grid
7 > wireframe2(val.W)
8 > wireframe2(val.M)
9 > contourplot2(val.W, xlim = 0:1, ylim = 0:1)
10 > contourplot2(val.M, xlim = 0:1, ylim = 0:1)
```

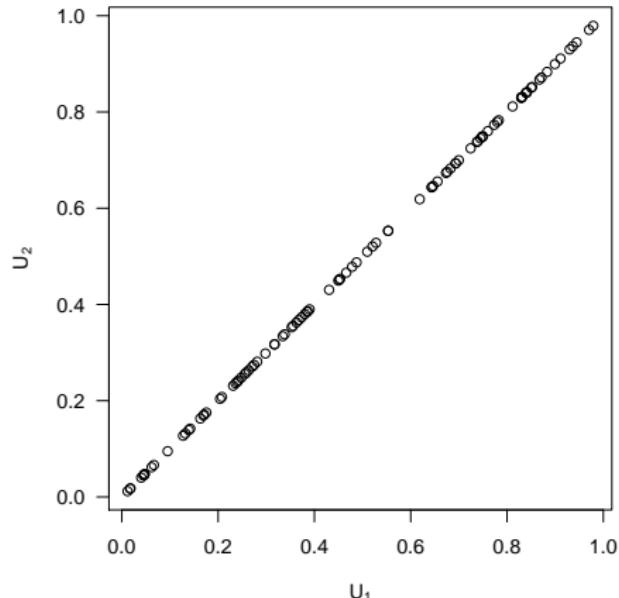
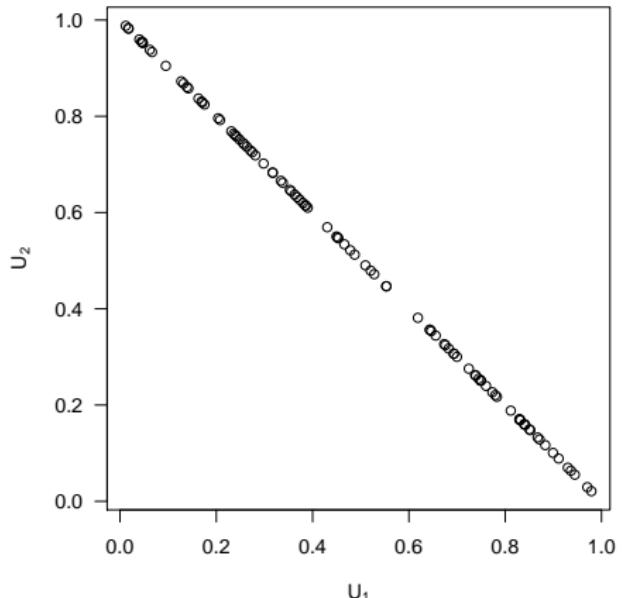


Figure 2.7 Scatter plot of $n = 100$ independent observations from W (left) and M (right).



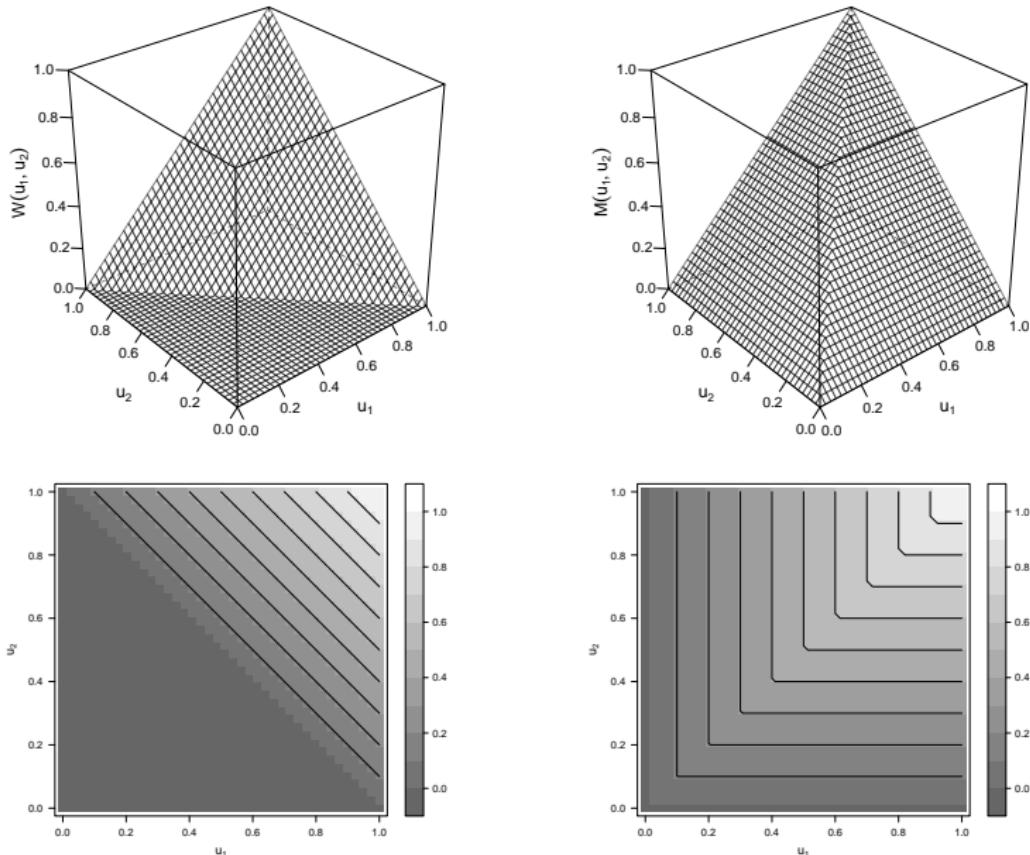


Figure 2.8 Wireframe (top) and contour plots (bottom) of W (left) and M (right).

- Neither W nor M are absolutely continuous; see Figure 2.7.
- Copulas (such as W and M) which put all probability mass on a set of (Lebesgue) measure 0 are called *singular*.
- Some copulas are not singular but have a *singular component*, that is, they put some probability mass in $(0, 1)$ on a set of (Lebesgue) measure 0.

Example 2.2.3 (Marshall–Olkin copulas)

- The bivariate *Marshall–Olkin* family of copulas is given by

$$C(u_1, u_2) = \min\{u_1 u_2^{1-\alpha_2}, u_1^{1-\alpha_1} u_2\}, \quad u_1, u_2 \in [0, 1], \quad (2.8)$$

with parameters $\alpha_1, \alpha_2 \in [0, 1]$; see Marshall and Olkin (1967).

- $(U_1, U_2) \sim C$ admits the *stochastic representation*

$$(U_1, U_2) = \left(\max\left\{V_1^{\frac{1}{1-\alpha_1}}, V_{12}^{\frac{1}{\alpha_1}}\right\}, \max\left\{V_2^{\frac{1}{1-\alpha_2}}, V_{12}^{\frac{1}{\alpha_2}}\right\} \right)$$

where $V_1, V_2, V_{12} \stackrel{\text{ind.}}{\sim} \text{U}(0, 1)$ and $v \in (0, 1)$, $v^{1/0} = 0$.

The stochastic representation can be used for sampling from C :

```
1 > ## A Marshall-Olkin copula
2 > C <- function(u, alpha)
3     pmin(u[,1] * u[,2]^(1 - alpha[2]), u[,1]^(1 - alpha[1]) * u[,2])
4 > alpha <- c(0.2, 0.8)
5 > val <- cbind(u12, "C(u[1],u[2])" = C(u12, alpha = alpha)) # append C values
6 > ## Generate data
7 > set.seed(712)
8 > V <- matrix(runif(1000 * 3), ncol = 3)
9 > U <- cbind(pmax(V[,1]^(1/(1 - alpha[1])), V[,3]^(1/alpha[1])),
10    pmax(V[,2]^(1/(1 - alpha[2])), V[,3]^(1/alpha[2])))
11 > ## Plots
12 > wireframe2(val)
13 > plot(U, xlab = quote(U[1]), ylab = quote(U[2]))
```

- Figure 2.9 shows the copula C (left) and a sample from it (right).
- Intuitively, if U_1 takes on a certain value u_1 , then, with a certain probability in $(0, 1)$, U_2 takes on the value $u_2 = u_1^{\alpha_1/\alpha_2}$ on the singular component; with the remaining probability, U_2 takes on a value on the strip defined by $U_1 = u_1$ off the singular component.

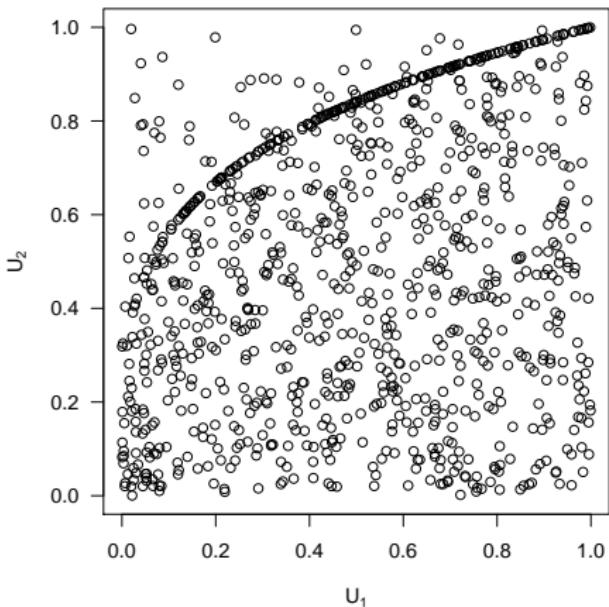
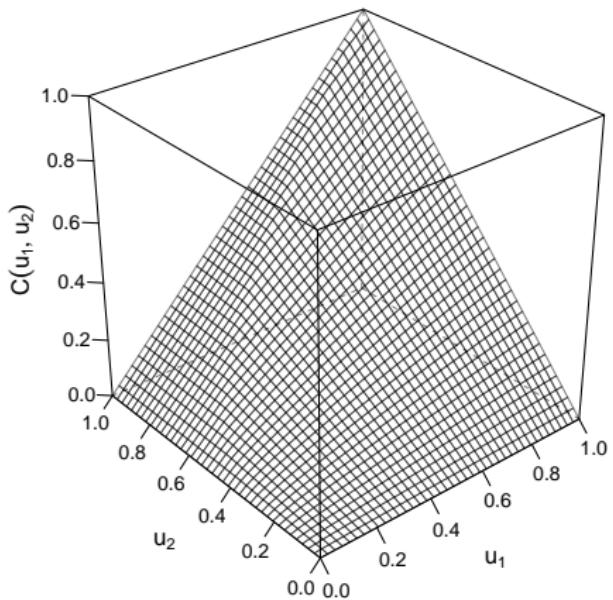


Figure 2.9 Plot of a Marshall–Olkin copula (left) and corresponding scatter plot (right) of $n = 1000$ independent observations. One can clearly see the singular component in the latter which is also reflected by a kink in the former (most easily spotted around $[0.4, 0.7] \times [0.8, 0.9]$).

2.3 Sklar's Theorem

- Sklar's Theorem is the central theorem of copula theory. It explains why copulas determine the dependence between the components of a random vector.
- An analytical proof can be found in Sklar (1996), a probabilistic one in Rüschedorf (2009).
- $\text{ran } F = \{F(x) : x \in \mathbb{R}\}$ denotes the range of a univariate df F and F^\leftarrow denotes the quantile function associated with F .
- If the main equation

$$H(\mathbf{x}) = C(F_1(x_1), \dots, F_d(x_d)), \quad \mathbf{x} \in \mathbb{R}^d,$$

holds for a df H with univariate margins F_1, \dots, F_d and if $\mathbf{X} \sim H$, we shall say that \mathbf{X} (or H) has copula C .

Here is the full result:

Theorem 2.3.1 (Sklar's Theorem)

- 1) For any d -dimensional df H with univariate margins F_1, \dots, F_d , there exists a d -dimensional copula C such that

$$H(\mathbf{x}) = C(F_1(x_1), \dots, F_d(x_d)), \quad \mathbf{x} \in \mathbb{R}^d. \quad (2.9)$$

The copula C is uniquely defined on $\prod_{j=1}^d \text{ran } F_j$ and is given by

$$C(\mathbf{u}) = H(F_1^\leftarrow(u_1), \dots, F_d^\leftarrow(u_d)), \quad \mathbf{u} \in \prod_{j=1}^d \text{ran } F_j. \quad (2.10)$$

- 2) Conversely, given a d -dimensional copula C and univariate dfs F_1, \dots, F_d , H defined by (2.9) is a d -dimensional df with margins F_1, \dots, F_d . □

Remark 2.3.2

- 1) By Sklar's Theorem, copulas are those functions which combine/couple the univariate marginal dfs F_1, \dots, F_d to form the d -dimensional df H .
- 2) If F_1, \dots, F_d in Part 1) are continuous, ran F_j contains $(0, 1)$ for all $j \in \{1, \dots, d\}$, so C in (2.10) is uniquely defined on $[0, 1]^d$. Consequently, (2.9) holds for only one d -dimensional copula. Statistical applications of copulas thus mostly concern the modeling of *continuous random vectors*, that is, random vectors with continuous marginal dfs.
- 3) By Sklar's Theorem, a random vector has a continuous df H if and only if it has continuous univariate marginal dfs F_1, \dots, F_d :
 - “ \Leftarrow ” previous point and the fact that any d -dimensional copula C is uniformly continuous on $[0, 1]^d$
 - “ \Rightarrow ” immediate consequence of the fact that, for any $j \in \{1, \dots, d\}$,
$$F_j(x_j) = H(\infty, \dots, \infty, x_j, \infty, \dots, \infty), x_j \in \mathbb{R}$$

4) From (2.9), it also follows that H is absolutely continuous if and only if C and F_1, \dots, F_d are absolutely continuous. In that case, the density h of H is

$$h(\mathbf{x}) = c(F_1(x_1), \dots, F_d(x_d)) \prod_{j=1}^d f_j(x_j), \quad \mathbf{x} \in \prod_{j=1}^d \text{ran } X_j,$$

where

- $\text{ran } X_j = \{x \in \mathbb{R} : \mathbb{P}(X_j \in (x - h, x]) > 0 \text{ for all } h > 0\}$ is the *range* of X_j ;
- f_j denotes the density of F_j ;
- c denotes the density of C .

In this case c can be recovered from h via

$$c(\mathbf{u}) = \frac{h(F_1^\leftarrow(u_1), \dots, F_d^\leftarrow(u_d))}{f_1(F_1^\leftarrow(u_1)) \cdot \dots \cdot f_d(F_d^\leftarrow(u_d))}, \quad \mathbf{u} \in (0, 1)^d. \quad (2.11)$$

- 5) The decomposition of H into F_1, \dots, F_d and C suggests to focus on two classes of dfs:
- i) the class of all d -dimensional dfs with given margins F_1, \dots, F_d known as a *Fréchet class*;
 - ii) the class of all multivariate dfs obtained from a given d -dimensional copula C known as *meta- C models*.

For example, a d -dimensional meta-II model consists of all multivariate df H such that $H(\mathbf{x}) = \prod_{j=1}^d F_j(x_j)$, $\mathbf{x} \in \mathbb{R}^d$ (a member of the Fréchet class obtained from F_1, \dots, F_d). □

- Statistical applications exploit mostly Part 1) of Sklar's Theorem. For example, once the margins and the copula are estimated, they can be coupled as in (2.9) to obtain the estimated multivariate df.
- Part 2) of Sklar's Theorem is often applied in risk management, finance or insurance for constructing flexible multivariate dfs with given margins. A

typical example is *stress testing* which is applied in settings in which one does not necessarily have enough multivariate observations to estimate C . Copulas can then be used to formulate dependence scenarios and to propagate their effects on risk measures of interest, for example, by means of simulation.

Example 2.3.3 (First part of Sklar's Theorem – decomposition)

Sklar's Theorem can be used to *create copulas from existing dfs*. Let H be the df of the bivariate standard normal $N_2(\mathbf{0}, P)$, where the correlation matrix P has off-diagonal entry $\rho = 0.7$. The corresponding *copula* can be evaluated via `pmvnorm()` of the R package `mvtnorm` of Genz et al. (2017):

```
1 > library(mvtnorm)
2 > d <- 2 # dimension
3 > rho <- 0.7 # off-diagonal entry of the correlation matrix P
4 > P <- matrix(rho, nrow = d, ncol = d) # build the correlation matrix P
5 > diag(P) <- 1
6 > set.seed(64)
7 > u <- runif(d) # generate a random evaluation point
8 > x <- qnorm(u)
9 > pmvnorm(upper = x, corr = P) # evaluate the copula C at u
```

```
[1] 0.04379748
attr(,"error")
[1] 1e-15
attr(,"msg")
[1] "Normal Completion"
```

This construction leads to the so-called *normal (or Gauss) copula* family. Its members are denoted by C_P^n and by C_ρ^n case, that is, when the correlation matrix P has equal off-diagonal elements ρ .

Objects representing C_P^n can be created using `normalCopula()`:

```
1 > nc <- normalCopula(rho) # normal copula (note the default dim = 2)
2 > pCopula(u, copula = nc) # value of the copula at u
```

```
[1] 0.04379748
```

Similarly, starting from a multivariate t distribution leads to the (*Student*) t *copula* family, parametrized by a correlation matrix P (referred to as a *scale matrix* or *dispersion matrix*) and a degrees of freedom parameter $\nu > 0$. Its members are denoted by $C_{P,\nu}^t$ in general and $C_{\rho,\nu}^t$ in the homogeneous case. Evaluating a t copula can, for instance, be done as follows:

```
1 > nu <- 3 # degrees of freedom
2 > x. <- qt(u, df = nu)
3 > pmvt(upper = x., corr = P, df = nu) # evaluate the t copula at u
```

```
[1] 0.04239823
attr(,"error")
[1] 1e-15
attr(,"msg")
[1] "Normal Completion"
```

Note that `pmvt()` from the R package `mvtnorm` can only be used with integer degrees of freedom:

```
1 > try( pmvt(upper = x., corr = P, df = 3.5) )
```

```
Error in pmvt(upper = x., corr = P, df = 3.5) : 'df' is not an integer
```

Objects representing t copulas can be created using `tCopula()`:

```
1 > tc <- tCopula(rho, dim = d, df = nu)
2 > pCopula(u, copula = tc) # value of the copula at u
```

```
[1] 0.04239823
```



Example 2.3.4 (Second part of Sklar's Theorem – composition)

Part 2) of Sklar's Theorem can be used to generate multivariate dfs. The following code shows how to create an R object representing a bivariate df with Clayton copula C_1^C , and $N(1, 4)$ and $\text{Exp}(3)$ margins:

```
1 > H.obj <- mvdc(claytonCopula(1), margins = c("norm", "exp"),
2                      paramMargins = list(list(mean = 1, sd = 2), list(rate = 3)))
```

The resulting object `H.obj` can then be used to evaluate the df and the corresponding density using `pMvdc()` and `dMvdc()`, respectively. For instance:

```
1 > set.seed(1979)
2 > z <- cbind(rnorm(5, mean = 1, sd = 2), rexp(5, rate = 3)) # evaluation points
3 > pMvdc(z, mvdc = H.obj) # values of the df at z
```

```
[1] 0.620039169 0.245159086 0.007856849 0.512835075 0.117870122
```

```
1 > dMvdc(z, mvdc = H.obj) # values of the corresponding density at z
```

```
[1] 0.10997504 0.03128093 0.19565908 0.17081495 0.40321721
```

A random sample from H can be generated using `rMvdc()`:

```
1 > set.seed(1975)
2 > X <- rMvdc(1000, mvdc = H.obj)
```

Figure 2.10 shows the corresponding scatter plot and contourplot:

```
1 > plot(X, cex = 0.5, xlab = quote(X[1]), ylab = quote(X[2]))
2 > contourplot2(H.obj, FUN = dMvdc, xlim = range(X[,1]), ylim = range(X[,2]),
3                 n.grid = 257)
```

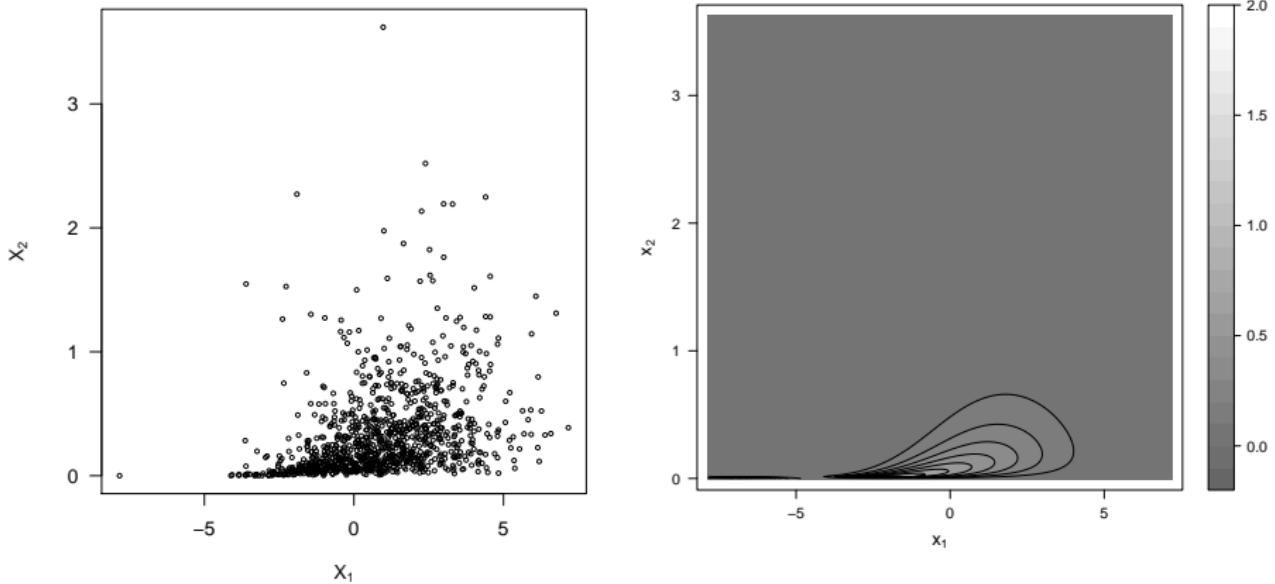


Figure 2.10 Scatter plot of $n = 1000$ independent observations from a bivariate distribution with copula C_1^C and margins $N(1, 4)$ and $\text{Exp}(3)$ (left) and contour plot of the corresponding density (right).



Example 2.3.5 (Risk aggregation)

- Application of Part 2) of Sklar's Theorem.
- Assume we have a continuous random vector $\mathbf{L} = (L_1, \dots, L_d)$, where L_j , $j \in \{1, \dots, d\}$, represents the non-negative loss in the j th business line of a financial firm over the next year.
- In risk aggregation one is interested in the aggregated loss

$$\mathbf{L}^+ = L_1 + \cdots + L_d$$

whose df F_{L^+} is known as *loss distribution*.

- A risk manager does not necessarily need to know F_{L^+} , but at least a risk measure $\varrho(L^+)$ based on F_{L^+} .
- One such measure is the so-called *value-at-risk* at *confidence level* $\alpha \in (0, 1)$, defined by

$$\text{VaR}_\alpha(L^+) = F_{L^+}^\leftarrow(\alpha),$$

that is, the α -quantile of F_{L^+} .

- The risk measure $\text{VaR}_\alpha(L^+)$ represents the smallest loss level which is exceeded with probability at most $1 - \alpha$; see McNeil et al. (2015, Chapter 2) for advantages and drawbacks of this risk measure.
- In applications, α is typically a given number close to 1, such as $\alpha = 0.99$ for market risk (over a 10 day time horizon) and $\alpha = 0.999$ for credit and operational risk (over a one year time horizon).
- Estimating $\text{VaR}_\alpha(L^+)$ is not an easy task. Based on historical losses (L_{i1}, \dots, L_{id}) , $i \in \{1, \dots, n\}$, seen as realizations of $\mathbf{L} = (L_1, \dots, L_d)$, one could consider the empirical α -quantile of the sample $L_i^+ = L_{i1} + \dots + L_{id}$, $i \in \{1, \dots, n\}$, as an estimate of $\text{VaR}_\alpha(L^+)$.
- However, the amount of available loss data is typically not sufficient to reliably estimate high quantiles such as $\text{VaR}_\alpha(L^+)$. Stress testing requires to estimate $\text{VaR}_\alpha(L^+)$ under specific copulas C of \mathbf{L} .
- For a target copula C , a first solution would be to simulate realizations of \mathbf{L} under C and the estimated marginal dfs, build the corresponding

simulated realizations of L^+ and utilize the corresponding empirical α -quantile as an estimate of $\text{VaR}_\alpha(L^+)$.

- Once $\text{VaR}_\alpha(L^+)$ is estimated for all the selected copulas (dependence scenarios), one option is to take as risk measure $\varrho(L^+)$ the maximal $\text{VaR}_\alpha(L^+)$, related to the worst case scenario.
- We present a different approach which does not require the estimation of the marginal loss distributions. We work with the matrix of losses $X = (L_{ij})$, $i \in \{1, \dots, n\}$, $j \in \{1, \dots, d\}$, and ask how we can create a matrix Y which contains, in each column (so for each marginal loss), the same numbers as X , but permuted within each column so that the rows of Y could be viewed as approximately being dependent according to C .
- Since we only permute the numbers within each column of X to get Y , the marginal losses are not changed. Only their dependence is adjusted to resemble that of C . To this end, we assume n to be sufficiently large.

To demonstrate the idea of transforming X into Y , we consider a three-dimensional example based on simulated loss data from Pareto, lognormal and loggamma marginal dfs. Let us start by generating X under this setup:

```
1 > ## Define parameters of the margins
2 > th <- 2.5 # Pareto parameter
3 > m <- 10 # mean of the lognormal
4 > v <- 20 # variance of the lognormal
5 > s <- 4 # shape of the underlying gamma
6 > r <- 5 # scale of the underlying gamma
7 > ## Define list of marginal dfs
8 > qF <- list(qPar = function(p) (1 - p)^(-1/th) - 1,
9 >                 qLN = function(p) qlnorm(p, meanlog = log(m)-log(1+v/m^2)/2,
10 >                               sdlog = sqrt(log(1+v/m^2))),
11 >                 qLG = function(p) exp(qgamma(p, shape = s, rate = r)))
12 > ## Generate the data
13 > set.seed(271) # for reproducibility
14 > X <- sapply(qF, function(mqf) mqf(runif(2500))) # (2500, 3)-matrix
```

- We consider homogeneous t copulas $C_{\rho,\nu}^t$ with $\nu = 3.5$ degrees of freedom and correlation parameter $\rho \in \{0, 0.1, \dots, 1\}$. Furthermore, we consider $\alpha \in \{0.001, 0.01, 0.05, 0.1, 0.2, \dots, 0.9, 0.95, 0.99, 0.999\}$.

- For convenience, we implement an auxiliary function which computes, for a given matrix X , (a vector of) confidence level(s) α , correlation parameter ρ and degrees of freedom $\nu = 3.5$ of a t copula, the empirical quantile estimator based on the row sums of Y , that is, a non-parametric estimator of $\text{VaR}_\alpha(L^+)$ under the dependence scenario $C_{\rho,3.5}^t$.
- The function is vectorized in α , thus allowing us to estimate $\text{VaR}_\alpha(L^+)$ for all α based on the same t copula sample. We also set the seed inside the auxiliary function; this is fine here as we do not call this function twice for the same value of ρ . Both these design decisions aim at reducing the variance across different ρ .

```

1 > #' @title Non-parametric VaR estimate under a t copula
2 > #' @param X loss matrix
3 > #' @param alpha confidence level
4 > #' @param rho correlation parameter of the t copula
5 > #' @param df degrees of freedom parameter of the t copula
6 > #' @return Non-parametric VaR estimate under the t copula (numeric)
7 > VaR <- function(X, alpha, rho, df = 3.5)
8 {
9   stopifnot(is.matrix(X), 0 <= rho, rho <= 1, length(rho) == 1,

```

```

10      0 < alpha, alpha < 1, length(alpha) >= 1)
11  n <- nrow(X) # sample size
12  d <- ncol(X) # dimension
13  ## Simulate from a t copula with d.o.f. parameter 3.5 and exchangeable
14  ## correlation matrix with off-diagonal entry rho. Also compute the
15  ## componentwise ranks.
16  ## Note: We can set the seed here as we can estimate VaR for all
17  ## confidence levels based on the same copula sample. We
18  ## even *should* set the seed here to minimize the variance
19  ## of the estimation and make the results more comparable.
20  set.seed(271)
21  U <- rCopula(n, copula = tCopula(rho, dim = d, df = df))
22  rk <- apply(U, 2, rank)
23  ## Componentwise reorder the data according to these ranks to
24  ## mimic the corresponding t copula dependence among the losses
25  Y <- sapply(1:d, function(j) sort(X[,j])[rk[,j]])
26  ## Build row sums to mimic a sample from the distribution of the
27  ## sum under the corresponding t copula.
28  S <- rowSums(Y)
29  ## Non-parametrically estimate VaR for all confidence levels alpha
30  ## Note: We use the mathematical definition ('type = 1') of a
31  ## quantile function here
32  quantile(S, probs = alpha, type = 1, names = FALSE)
33 }

```

- Now build a grid of the aforementioned α and ρ values and compute $\text{VaR}_\alpha(L^+)$ for all combinations of these two inputs.

```

1 > alpha <- c(0.001, 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5,
2   0.6, 0.7, 0.8, 0.9, 0.95, 0.99, 0.999) # confidence levels
3 > rho <- seq(0, 1, by = 0.1) # parameter of the homogeneous t copula
4 > grid <- expand.grid("alpha" = alpha, "rho" = rho)[,2:1] # build a grid
5 > VaR.fit <- sapply(rho, function(r)
6   VaR(X, alpha = alpha, rho = r)) # (alpha, rho)
7 > res <- cbind(grid, "VaR[alpha](L^+)" = as.vector(VaR.fit))

```

- The left-hand side of Figure 2.11 shows the non-parametrically estimated $\text{VaR}_\alpha(L^+)$.
- For larger α , the (empirical) estimator is based on less and less data and thus affected by a larger variance.
- For large (small) α , $\text{VaR}_\alpha(L^+)$ seems increasing (decreasing) in ρ .
- The right-hand side of Figure 2.11 shows, for each considered α , the maximal $\text{VaR}_\alpha(L^+)$ over all considered ρ (and thus the worst $\text{VaR}_\alpha(L^+)$ over all the considered $C_{\rho,3.5}^t$ copulas for the given marginal dfs).

- It also includes the *worst value-at-risk*, that is, the largest $\text{VaR}_\alpha(L^+)$ for our given marginal dfs under all possible copulas (obtained from the adaptive rearrangement algorithm (ARA) of the R package qrmtools of Hofert and Hornik (2017)).

```

1 > wireframe2(res)
2 > library(qrmtools)
3 > worst.VaR <- sapply(alpha, function(a) mean(ARA(a, qF = qF)$bounds))
4 > plot(alpha, worst.VaR, type = "b", col = 2,
5       xlab = quote(alpha), ylab = quote(VaR[alpha](L^'+)),
6       ylim = range(VaR.fit, worst.VaR)) # computed with the ARA
7 > lines(alpha, apply(VaR.fit, 1, max), type = "b", col = 1) # simulated
8 > legend("topleft", bty = "n", lty = rep(1, 2), col = 2:1,
9       legend = c(expression("Worst"~VaR[alpha]~"according to ARA()"),
10                  expression("Worst"~VaR[alpha]~"under"~t[3.5]~"copulas")))

```

- As we can see from the right-hand side of Figure 2.11, for no α does the worst $\text{VaR}_\alpha(L^+)$ over all considered $C_{\rho,3.5}^t$ attain the worst $\text{VaR}_\alpha(L^+)$ over all copulas, not even for $\rho = 1$ (which results in $C_{\rho,3.5}^t$ being the comonotone copula M).

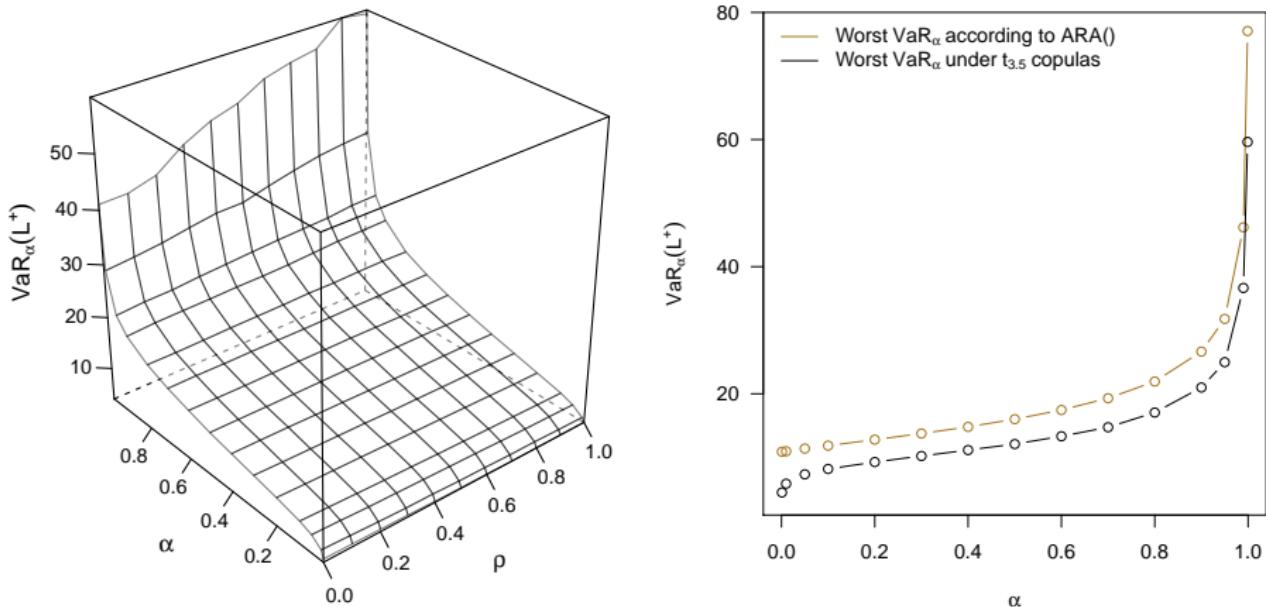


Figure 2.11 Non-parametrically estimated $\text{VaR}_\alpha(L^+)$ as a function of ρ and α based on rearranging the given Pareto, lognormal and loggamma marginal losses according to the component ranks of a sample from a homogeneous $C_{\rho,3.5}^t$ copula (left). Worst $\text{VaR}_\alpha(L^+)$ over all considered t copulas $C_{\rho,3.5}^t$ and approximate worst $\text{VaR}_\alpha(L^+)$ over all possible copulas according to the ARA (right).

- It is a common misbelief that maximal correlation leads to maximal $\text{VaR}_\alpha(L^+)$. The reason why this argument fails is that $\text{VaR}_\alpha(L^+)$ can be superadditive; see McNeil et al. (2015, Section 2.3.5).
- The left-hand side of Figure 2.12 shows a scatter-plot matrix of an approximate sample from this worst $\text{VaR}_\alpha(L^+)$ copula for $\alpha = 0.99$.
- The right-hand side of Figure 2.12 shows a scatter plot of a pseudo-sample from the worst $\text{VaR}_\alpha(L^+)$ copula for $d = 2$ (corresponding to the first two of the three considered marginal dfs); as we can see, it corresponds to the countermonotone copula W .

```

1 > ## Computing worst VaR in the three-dimensional case
2 > wVaR <- ARA(0.99, qF = qF) # compute worst VaR (bounds)
3 > X <- wVaR[["X.rearranged"]]\$up # extract rearranged matrix (upper bound)
4 > U <- pobs(X) # compute pseudo-observations
5 > pairs2(U) # approx. sample of a copula leading to worst VaR for our marg. dfs
6 > ## Computing worst VaR in the bivariate case
7 > wVaR. <- ARA(0.99, qF = qF[1:2]) # compute worst VaR (bounds)
8 > X. <- wVaR.[["X.rearranged"]]\$up # extract rearranged matrix (upper bound)
9 > U. <- pobs(X.) # compute pseudo-observations
10 > plot(U., xlab = quote(U[1]), ylab = quote(U[2]))

```

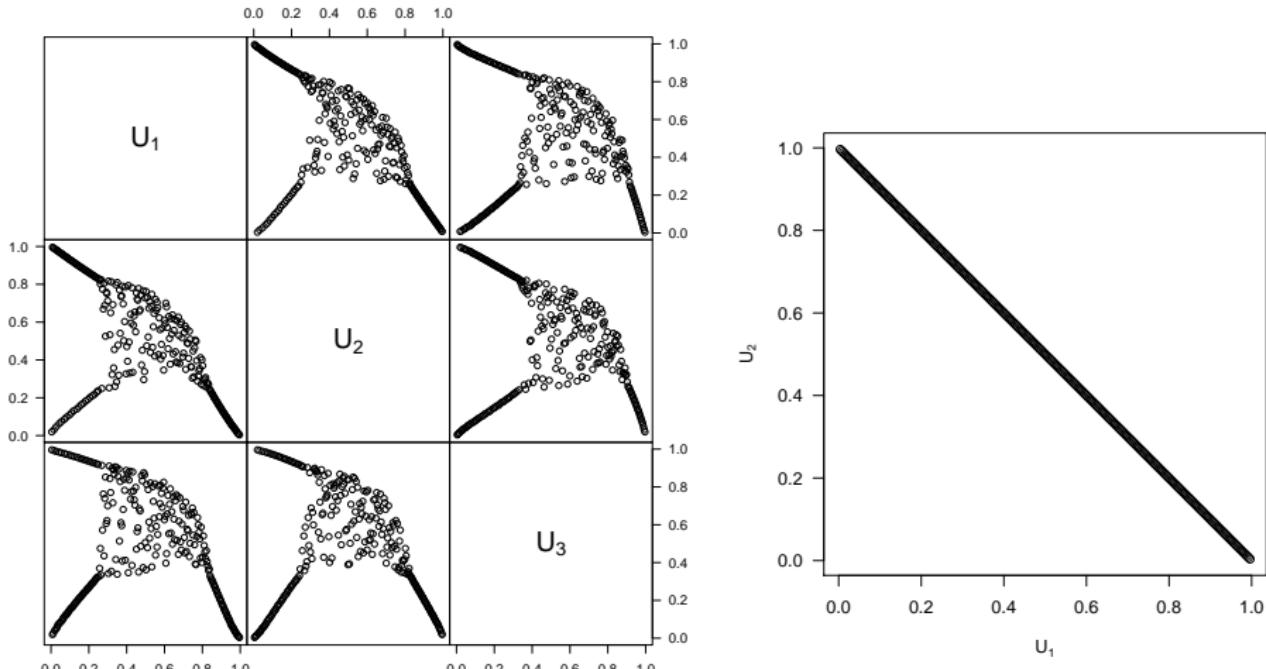


Figure 2.12 Approximate sample from the copula leading to the worst $\text{VaR}_\alpha(L^+)$ for the given Pareto, lognormal and loggamma marginal dfs (left) and for the bivariate restriction of the problem for the given Pareto and lognormal marginal dfs (right).

2.4 The invariance principle

Lemma 2.4.1 (Stochastic analog of Sklar's Theorem)

Let \mathbf{X} be a d -dimensional random vector with continuous univariate marginal dfs F_1, \dots, F_d . Then, \mathbf{X} has copula C if and only if $(F_1(X_1), \dots, F_d(X_d)) \sim C$. \square

- The necessity (“ \Rightarrow ”) in Lemma 2.4.1 allows us to construct a random vector \mathbf{U} distributed according to the underlying copula C . Since we can recover C from H in this sense, this is a stochastic analog of Part 1) of Sklar's Theorem.
- The sufficiency in Lemma 2.4.1 (“ \Leftarrow ”) allows us to construct a random vector $\mathbf{X} \sim H$ with copula C , namely via

$$(F_1^\leftarrow(U_1), \dots, F_d^\leftarrow(U_d)) \sim H.$$

This is a stochastic analog of Part 2) of Sklar's Theorem.

Example 2.4.2 (Sampling from a normal or t copula)

As an application of Lemma 2.4.1, we can sample from the t copula family using the `mvtnorm` package:

```
1 > n <- 1000 # sample size
2 > d <- 2 # dimension
3 > rho <- 0.7 # off-diagonal entry in the correlation matrix P
4 > P <- matrix(rho, nrow = d, ncol = d) # build the correlation matrix P
5 > diag(P) <- 1
6 > nu <- 3.5 # degrees of freedom
7 > set.seed(271)
8 > X <- rmvt(n, sigma = P, df = nu) # n ind. multivariate t observations
9 > U <- pt(X, df = nu) # n ind. realizations from the corresponding copula
```

`rCopula()` (for the t copula family) is just a wrapper:

```
1 > set.seed(271)
2 > U. <- rCopula(n, tCopula(rho, dim = d, df = nu))
3 > stopifnot(all.equal(U, U.)) # test of (numerical) equality
```

The previous numerical equality is also confirmed by the scatter plots shown in Figure 2.13:

```
1 > plot(U., xlab = quote(U[1]), ylab = quote(U[2]))
2 > plot(U, xlab = quote(U[1]), ylab = quote(U[2]))
```

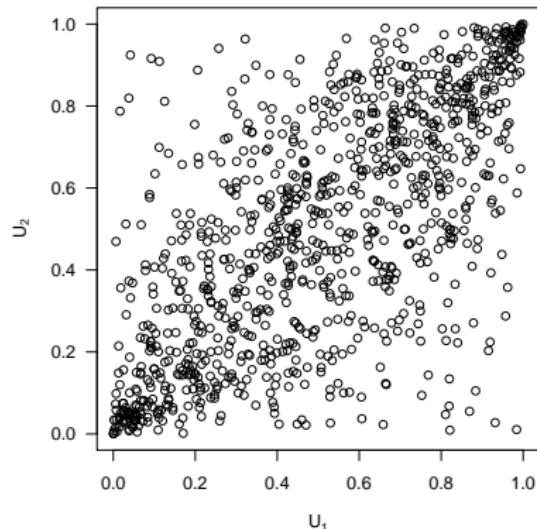
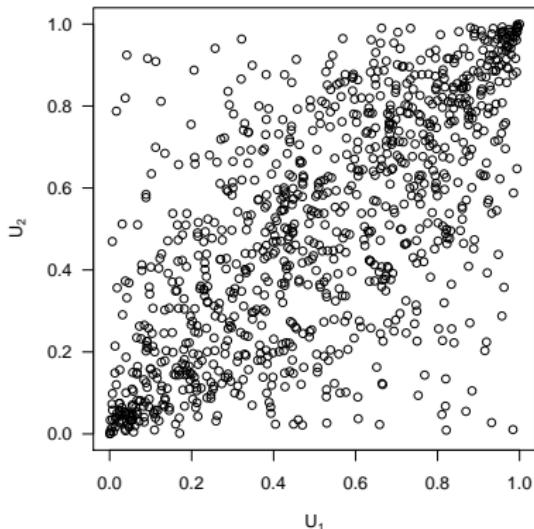


Figure 2.13 Scatter plots of $n = 1000$ independent observations generated from $C_{\rho,\nu}^t$ for $\rho = 0.7$ and $\nu = 3.5$ manually (left) and via `rCopula()` (right).

Example 2.4.3 (From a multivariate t distribution to a t copula to a meta- t model)

- Based on the sample from $C_{\rho,\nu}^t$, $\rho = 0.7$, $\nu = 3.5$ as generated in Example 2.4.2, we can use Lemma 2.4.1 to obtain observations from meta- t models.
- To illustrate this, we map a multivariate t sample to a sample from $C_{\rho,\nu}^t$ and then to a multivariate distribution with $C_{\rho,\nu}^t$ and $N(0, 1)$ margins.
- If we had used **t quantile functions** with degrees of freedom different than ν instead, we would have obtained a sample from a bivariate distribution that, as the bivariate t distribution, has a t copula and univariate t margins, but which is **more flexible** in the sense of allowing for different degrees of freedom for each margin and the t copula.

```
1 > Y <- qnorm(U) # transform U (t copula) to normal margins
2 > ind <- c(A = 725, B = 351, C = 734) # use 'plot(X); identify(X)' to find them
3 > ## Plot function highlighting A, B, C
4 > plotABC <- function(x, col = adjustcolor("black", 1/2), pch = 19, ...)
5 {
```

```

6     cols <- adjustcolor(c("red", "blue", "magenta"), offset = -c(1,1,1,1.5)/4)
7     par(pty = "s")
8     plot(x, col = col, asp = 1,...)
9     xy <- x[ind, , drop = FALSE]
10    points(xy, pch = pch, col = cols)
11    text(xy, label = names(ind), adj = c(0.5, -0.6), col = cols, font = 2)
12  }
13 > ## Scatter plot of observations from the multivariate t distribution
14 > plotABC(X, xlab = quote(X[1]), ylab = quote(X[2]))
15 > ## Scatter plot of observations from the corresponding t copula
16 > plotABC(U, xlab = quote(U[1]), ylab = quote(U[2]))
17 > ## Scatter plot of observations from the meta-t distribution
18 > plotABC(Y, xlab = quote(Y[1]), ylab = quote(Y[2]))

```

- The colorized points show how the strictly increasing transformations act on the sample: the relative locations of the points A, B and C are unchanged in the sense that their order according to their first (respectively, second) coordinate remains the same.
- In other words, the componentwise ranks of the points were not affected by the two transformations. This is not a surprise since ranks are invariant to strictly increasing transformations.

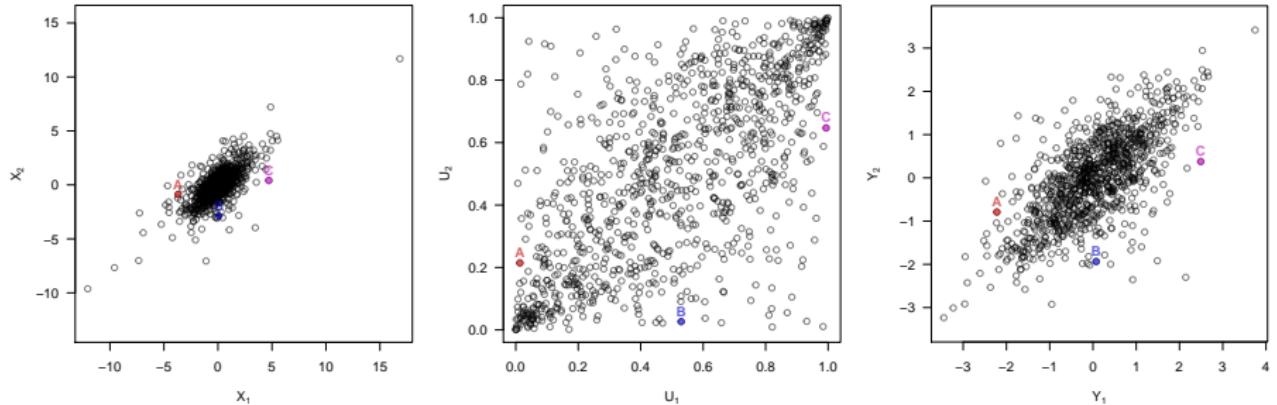


Figure 2.14 Scatter plots of $n = 1000$ independent observations from a multivariate t distribution (left), the corresponding (probability transformed) sample from the t copula (middle) and the corresponding marginally normally distributed (quantile transformed) meta- t sample (right).



We now quickly provide (the) two sampling algorithms which follow from Lemma 2.4.1. Algorithm 2.4.4 can be used to sample *implicit copulas* (that is, copulas implicitly given by Sklar's Theorem) such as the normal and t copulas (as done in Example 2.4.2).

Algorithm 2.4.4 (Simulation of implicit copulas)

- 1) Sample $\mathbf{X} \sim H$, where H is a d -dimensional df with continuous margins F_1, \dots, F_d .
- 2) Return $\mathbf{U} = (F_1(X_1), \dots, F_d(X_d))$. □

Algorithm 2.4.5 can be used to sample meta- C models (see Example 2.4.3).

Algorithm 2.4.5 (Simulation of meta- C models)

- 1) Sample $\mathbf{U} \sim C$.
- 2) Return $\mathbf{X} = (F_1^\leftarrow(U_1), \dots, F_d^\leftarrow(U_d))$. □

Remark 2.4.6 (Stochastic interpretations of W and M)

Lemma 2.4.1 can also be used to show the following stochastic interpretations of the Fréchet–Höffding bounds:

- 1) Let (X_1, X_2) be a continuous random vector. The copula of (X_1, X_2) is the lower Fréchet–Höffding bound W if and only if X_2 is (almost surely) a strictly decreasing function of X_1 . In this case, X_1 and X_2 are said to be *perfectly negatively dependent* and are called *countermonotone*.
- 2) Let $d \geq 2$ and $\mathbf{X} = (X_1, \dots, X_d)$ be a continuous random vector. The copula of \mathbf{X} is the upper Fréchet–Höffding bound M if and only if X_2, \dots, X_d are (almost surely) strictly increasing functions of X_1 . In this case, X_1, \dots, X_d are said to be *perfectly positively dependent* and are called *comonotone*.

Note that these results apply, in particular, to the random vectors $(U, 1 - U)$ and (U, \dots, U) , with $U \sim \text{U}(0, 1)$. □

We now present the invariance principle which states that copulas are invariant under strictly increasing transformations on the ranges of the underlying random variables.

Theorem 2.4.7 (Invariance principle)

Let $\mathbf{X} \sim H$ with continuous univariate margins F_1, \dots, F_d and copula C . If, for any $j \in \{1, \dots, d\}$, T_j is a strictly increasing transformation on $\text{ran } X_j$, then $(T_1(X_1), \dots, T_d(X_d))$ (also) has copula C . \square

- The invariance principle immediately applies to $\mathbf{X} \sim H$ with $T_j = F_j$, $j \in \{1, \dots, d\}$.
- It also applies to $\mathbf{U} = (U_1, \dots, U_d)$ with quantile functions $F_1^\leftarrow, \dots, F_d^\leftarrow$ which are strictly increasing on $(0, 1)$.
- Figure 2.14 (three colored points) can be seen as an illustration of these two settings.

Example 2.4.8 (Verifying the invariance principle)

- Let Φ_P denote the df of the bivariate standard normal whose correlation matrix P has off-diagonal element ρ and consider $(X_1, X_2) \sim \Phi_P$.
- Part 1) of Sklar's Theorem implies that the copula C of (X_1, X_2) is the normal copula C_ρ^n .
- Consider $T_1(x) = \exp(x)$ and $T_2(x) = 1/(1 + \exp(-x))$, $x \in \mathbb{R}$, which are both strictly increasing on $\mathbb{R} = \text{ran } X_j$, $j \in \{1, 2\}$.
- Analytical verification:**

- The df \tilde{H} of $(T_1(X_1), T_2(X_2))$ is

$$\tilde{H}(x_1, x_2) = \mathbb{P}(T_1(X_1) \leq x_1, T_2(X_2) \leq x_2)$$

$$= \mathbb{P}(X_1 \leq T_1^{-1}(x_1), X_2 \leq T_2^{-1}(x_2))$$

$$= \Phi_P\left(\log x_1, -\log\left(\frac{1-x_2}{x_2}\right)\right), \quad (x_1, x_2) \in (0, \infty) \times (0, 1).$$

- ▶ The corresponding marginals $\tilde{F}_1(x_1) = \Phi(\log x_1)$, $x_1 > 0$ (obtained for $x_2 \uparrow 1$) and $\tilde{F}_2(x_2) = \Phi(-\log((1 - x_2)/x_2))$, $x_2 \in (0, 1)$ (obtained for $x_1 \uparrow \infty$) have quantile functions $\tilde{F}_1^{-1}(u) = \exp(\Phi^{-1}(u))$ and $\tilde{F}_2^{-1}(u) = 1/(1 + \exp(-\Phi^{-1}(u)))$, $u \in (0, 1)$, respectively.
- ▶ By Sklar's Theorem, the copula of $(T_1(X_1), T_2(X_2))$ is thus

$$\begin{aligned}\tilde{C}(u_1, u_2) &= \tilde{H}(\tilde{F}_1^{-1}(u_1), \tilde{F}_2^{-1}(u_2)) \\ &= \Phi_P\left(\log(\exp(\Phi^{-1}(u_1))), -\log\left(\frac{1 - 1/(1 + \exp(-\Phi^{-1}(u_2)))}{1/(1 + \exp(-\Phi^{-1}(u_2)))}\right)\right) \\ &= \Phi_P\left(\Phi^{-1}(u_1), -\log(\exp(-\Phi^{-1}(u_2)))\right) = \Phi_P(\Phi^{-1}(u_1), \Phi^{-1}(u_2)) \\ &= C(u_1, u_2), \quad (u_1, u_2) \in (0, 1)^2,\end{aligned}$$

that is, the (same) copula C as the one of (X_1, X_2) .

- ▶ We can also verify the equality of \tilde{C} and C at selected points in R:

```
1 > rho <- 0.6
2 > P <- matrix(c(1, rho, rho, 1), ncol = 2) # the correlation matrix
```

```

3 > C <- function(u) pCopula(u, copula = normalCopula(rho)) # normal copula
4 > Htilde <- function(x)
5     apply(cbind(log(x[,1]), -log((1-x[,2])/x[,2])), 1, function(x.)
6         pmvnorm(upper = x., corr = P))
7 > qF1tilde <- function(u) exp(qnorm(u))
8 > qF2tilde <- function(u) 1/(1+exp(-qnorm(u)))
9 > Ctilde <- function(u) Htilde(cbind(qF1tilde(u[,1]), qF2tilde(u[,2])))
10 > set.seed(31)
11 > u <- matrix(runif(5 * 2), ncol = 2) # 5 random evaluation points
12 > stopifnot(all.equal(Ctilde(u), C(u)))

```

- ▶ The general proof follows along the same lines.
- **Stochastic verification:**
 - ▶ Sample $(T_1(X_1), T_2(X_2))$. Applying the dfs \tilde{F}_j to $T_j(X_j)$, $j \in \{1, 2\}$, gives us a sample from the copula \tilde{C} of $(T_1(X_1), T_2(X_2))$ (see Lemma 2.4.1), which we can then compare (in distribution) to a sample of C .
 - ▶ We can check this here with exact (numerical) equality (not just in distribution):

```

1 > set.seed(721)
2 > X <- rmvnorm(1000, mean = c(0,0), sigma = P) # sample from  $N(0, P)$ 
3 > ## 'Sample' the copula of  $X$  directly
4 > U <- pnorm(X)
5 > ## Transform the sample  $X$  componentwise
6 > TX <- cbind(exp(X[,1]), plogis(X[,2])) # note:  $plogis(x) = 1/(1+exp(-x))$ 
7 > ## Apply the marginal dfs to get a sample from the copula of  $TX$ 
8 > ## Note:  $qlogis(p) == logit(p) == log(p/(1-p))$ 
9 > V <- cbind(pnorm(log(TX[,1])), pnorm(qlogis(TX[,2])))
10 > stopifnot(all.equal(V, U)) # => the samples of the two copulas are the same

```



2.5 Survival copulas and copula symmetries

- The *survival function* \bar{F} corresponding to a univariate df $F(x) = \mathbb{P}(X \leq x)$, $x \in \mathbb{R}$, is defined by

$$\bar{F}(x) = \mathbb{P}(X > x), \quad x \in \mathbb{R},$$

and thus satisfies $\bar{F}(x) = 1 - F(x)$, $x \in \mathbb{R}$.

- Similarly, the *multivariate survival function* \bar{H} corresponding to a d -dimensional df $H(\mathbf{x}) = \mathbb{P}(\mathbf{X} \leq \mathbf{x})$ is defined by

$$\bar{H}(\mathbf{x}) = \mathbb{P}(\mathbf{X} > \mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d;$$

note that if $d > 1$, $\bar{H}(\mathbf{x}) \neq 1 - H(\mathbf{x})$ in general.

- The *univariate margins* of \bar{H} or *marginal survival functions* \bar{F}_j of $\mathbf{X} \sim H$ can either be obtained from F_j or from \bar{H} via

$$\bar{F}_j(x_j) = \bar{H}(-\infty, \dots, -\infty, x_j, -\infty, \dots, -\infty), \quad x_j \in \mathbb{R}.$$

- Sklar's Theorem can equally well be formulated in terms of survival functions.
- This is used to construct certain copula classes (the marginal survival functions are often concentrated on $[0, \infty)$).

Theorem 2.5.1 (Sklar's Theorem for survival functions)

- 1) For any d -dimensional survival function \bar{H} with margins $\bar{F}_1, \dots, \bar{F}_d$, there exists a copula \bar{C} such that

$$\bar{H}(\mathbf{x}) = \bar{C}(\bar{F}_1(x_1), \dots, \bar{F}_d(x_d)), \quad \mathbf{x} \in \mathbb{R}^d. \quad (2.12)$$

The copula \bar{C} is uniquely defined on $\prod_{j=1}^d \text{ran } \bar{F}_j$ and there given by

$$\bar{C}(\mathbf{u}) = \bar{H}(F_1^\leftarrow(1 - u_1), \dots, F_d^\leftarrow(1 - u_d)). \quad (2.13)$$

- 2) Conversely, given any d -dimensional copula \bar{C} and univariate survival functions $\bar{F}_1, \dots, \bar{F}_d$, \bar{H} defined by (2.12) is a survival function with margins $\bar{F}_1, \dots, \bar{F}_d$. □

- \bar{C} is called the *survival copula* of \mathbf{X} (or H).
- Note that \bar{C} is a copula (and thus a df), whereas neither \bar{H} nor $\bar{F}_1, \dots, \bar{F}_d$ are dfs.

- Bivariate Marshall–Olkin copulas as in (2.8) were originally constructed as survival copulas of lifetimes of the form

$$X_1 = \min\{Z_1, Z_{12}\} \quad \text{and} \quad X_2 = \min\{Z_2, Z_{12}\},$$

where $Z_1 \sim \text{Exp}(\lambda_1)$, $Z_2 \sim \text{Exp}(\lambda_2)$ and $Z_{12} \sim \text{Exp}(\lambda_{12})$ are independent (arrival times of individual/joint fatal shock to a system). Using the parametrization $\alpha_j = \lambda_j / (\lambda_j + \lambda_{12})$, $j \in \{1, 2\}$, and including the boundary cases $\alpha_j \in \{0, 1\}$, $j \in \{1, 2\}$, Marshall–Olkin copulas arise.

- Is there a stochastic representation of a random vector following \bar{C} given a random vector following C ?
- Can \bar{C} be expressed in terms of C ?
- The following proposition provides answers. Part 1) is a consequence of Sklar's Theorem for survival functions; Part 2) follows from the inclusion–exclusion principle; and Part 3) is a direct consequence of Part 2).

Proposition 2.5.2 (Properties of survival copulas)

Let C be a copula and let $\mathbf{U} \sim C$. Then:

- 1) $\mathbf{1} - \mathbf{U} \sim \bar{C}$, that is, $\mathbf{1} - \mathbf{U} = (1 - U_1, \dots, 1 - U_d)$ is a random vector whose df is the survival copula \bar{C} corresponding to C .
- 2) The survival copula \bar{C} itself can be computed from C via

$$\bar{C}(\mathbf{u}) = \sum_{J \subseteq \{1, \dots, d\}} (-1)^{|J|} C((1 - u_1)^{1(1 \in J)}, \dots, (1 - u_d)^{1(d \in J)}), \quad (2.14)$$

where the sum extends over all 2^d subsets J of $\{1, \dots, d\}$, $|J|$ denotes the number of elements of J and $1(j \in J)$ is the indicator of $j \in J$. For $d = 2$, $\bar{C}(\mathbf{u}) = 1 - (1 - u_1) - (1 - u_2) + C(1 - u_1, 1 - u_2) = -1 + u_1 + u_2 + C(1 - u_1, 1 - u_2)$.

- 3) If C admits a density, so does \bar{C} and the density \bar{c} of \bar{C} is given by

$$\bar{c}(\mathbf{u}) = c(1 - u_1, \dots, 1 - u_d), \quad \mathbf{u} \in (0, 1)^d.$$

□

Example 2.5.3 (Survival copulas)

Consider $C = C_2^C$, $\mathbf{U} \sim C$ and exploit the fact that $\mathbf{V} \sim \bar{C}$ can be immediately obtained from $\mathbf{U} \sim C$ using $\mathbf{V} = \mathbf{1} - \mathbf{U}$:

```
1 > cc <- claytonCopula(2)
2 > set.seed(271)
3 > U <- rCopula(1000, copula = cc) # sample from the Clayton copula
4 > V <- 1 - U # sample from the survival Clayton copula
5 > plot(U, xlab = quote(U[1]), ylab = quote(U[2])) # scatter plot
6 > plot(V, xlab = quote(V[1]), ylab = quote(V[2])) # for the survival copula
```

- For any $d \geq 2$, a sample from \bar{C} can be obtained from a sample from C by point reflection with respect to $\mathbf{1}/2 = (1/2, \dots, 1/2)$.
- Note that, summing up the terms in (2.14) (even when sorted according to their signs) can be affected by round-off errors and may result in a number outside $[0, 1]$. Also, it is prohibitive for large d .
- Objects representing survival copulas can be created with `rotCopula()`:

```
1 > wireframe2(cc,           FUN = dCopula, delta = 0.025)
2 > wireframe2(rotCopula(cc), FUN = dCopula, delta = 0.025)
```

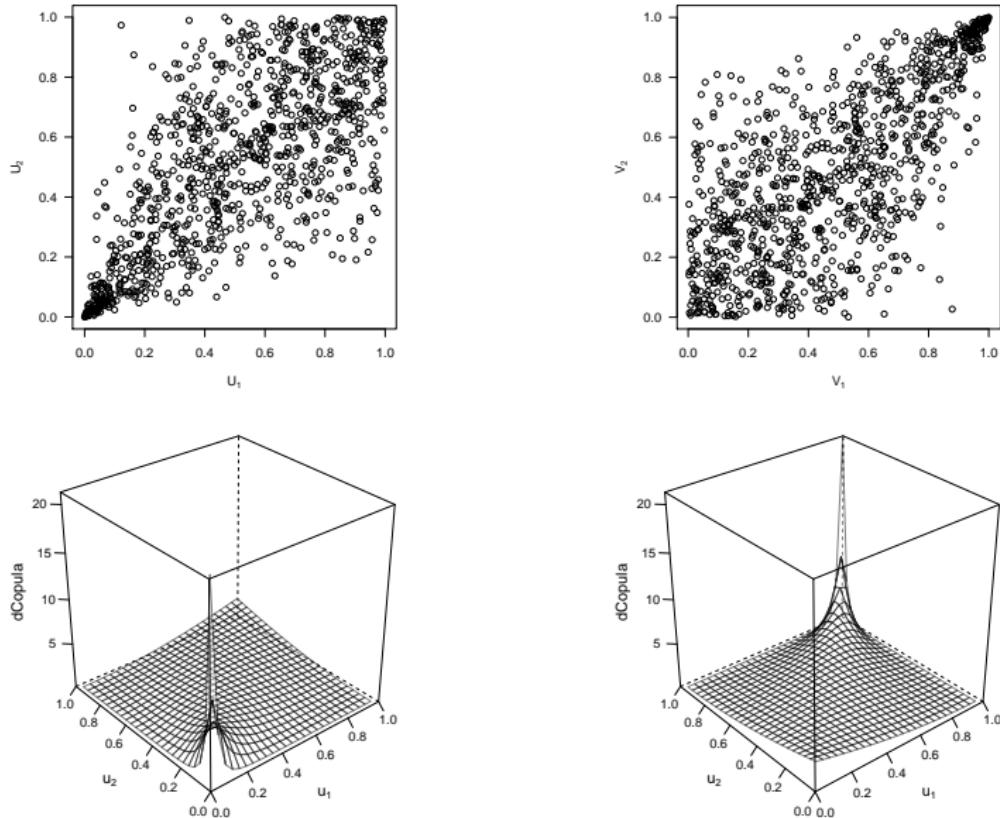


Figure 2.15 Scatter plots (top) and wireframe plots of the densities (bottom) of a Clayton copula C_θ^C (left) and survival Clayton copula \bar{C}_θ^C (right) with parameter $\theta = 2$. \square

Definition 2.5.4 (Radial symmetry, exchangeability)

- 1) A random vector \mathbf{X} is called *radially symmetric* (for $d = 1$ simply *symmetric*) about $\mathbf{a} \in \mathbb{R}^d$ if $\mathbf{X} - \mathbf{a} \stackrel{d}{=} \mathbf{a} - \mathbf{X}$.
- 2) The random vector \mathbf{X} is called *exchangeable* if $(X_{j_1}, \dots, X_{j_d}) \stackrel{d}{=} (X_1, \dots, X_d)$ for all permutations (j_1, \dots, j_d) of $\{1, \dots, d\}$. \square

Proposition 2.5.5 (Characterization of radial symmetry and exchangeability)

Let $\mathbf{X} \sim H$ with continuous margins F_1, \dots, F_d and copula C .

- 1) If X_j is symmetric about a_j , $j \in \{1, \dots, d\}$, then \mathbf{X} is radially symmetric about \mathbf{a} if and only if $C = \bar{C}$.
- 2) \mathbf{X} is exchangeable if and only if $F_1 = \dots = F_d$ and $C(u_{j_1}, \dots, u_{j_d}) = C(u_1, \dots, u_d)$ for all permutations (j_1, \dots, j_d) of $\{1, \dots, d\}$. \square

- If $\bar{C} = C$, we call C *radially symmetric*.
- If $C(u_{j_1}, \dots, u_{j_d}) = C(u_1, \dots, u_d)$ for all permutations (j_1, \dots, j_d) of $\{1, \dots, d\}$, we call C *exchangeable*.
- **Examples:**
 - ▶ Radially symmetric and exchangeable:
 W , Π , M and homogeneous normal and t copulas
 - ▶ Not radially symmetric but exchangeable:
Gumbel–Hougaard (see next example)
 - ▶ Radially symmetric but not exchangeable:
Consider (2.16) in Example 2.6.3 below with $h_1(u) = \sin(\pi u)/\pi$ and $h_2(u) = u(1 - u)$, $u \in [0, 1]$ to.
 - ▶ Neither radially symmetric nor exchangeable:
Marshall–Olkin copulas with distinct parameters $\alpha_1, \alpha_2 \in (0, 1)$; see Example 2.2.3

- Geometrically, if its density exists, radial symmetry of a copula means symmetry of the density with respect to $\mathbf{1}/2 = (1/2, \dots, 1/2)$.
- Exchangeability can be inferred from the symmetry of the density (or of the copula) with respect to the main diagonal.
- For $d = 2$, radial symmetry and exchangeability can be assessed from a scatter plot provided a sufficiently large sample is available.

Example 2.5.6 (Visually assessing radial symmetry and exchangeability)

Figure 2.16 shows contour plots of the densities $c_{\rho,\nu}^t$ and c_θ^{GH} of a t copula $C_{\rho,\nu}^t$ with correlation parameter $\rho = 0.7$ and degrees of freedom $\nu = 3.5$ (left) and a Gumbel–Hougaard copula C_θ^{GH} with parameter $\theta = 2$ (right):

```

1 > contourplot2(tCopula(0.7, df = 3.5), FUN = dCopula, n.grid = 64, lwd = 1/2)
2 > contourplot2(gumbelCopula(2),           FUN = dCopula, n.grid = 64, lwd = 1/4,
3               pretty = FALSE, cuts = 42,
4               col.regions = gray(seq(0.5, 1, length.out = 128)))

```

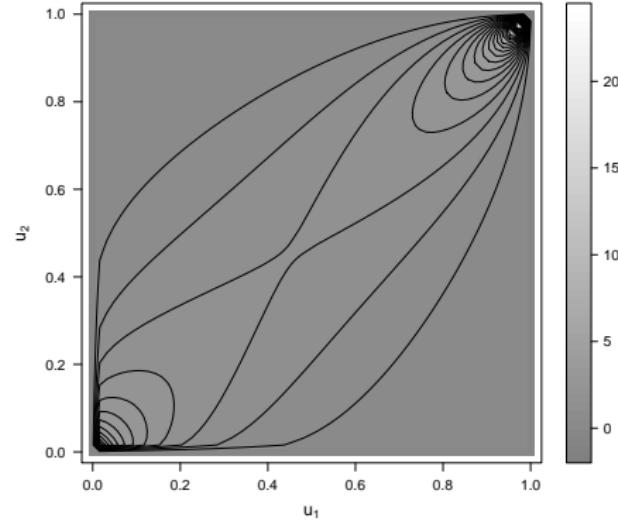
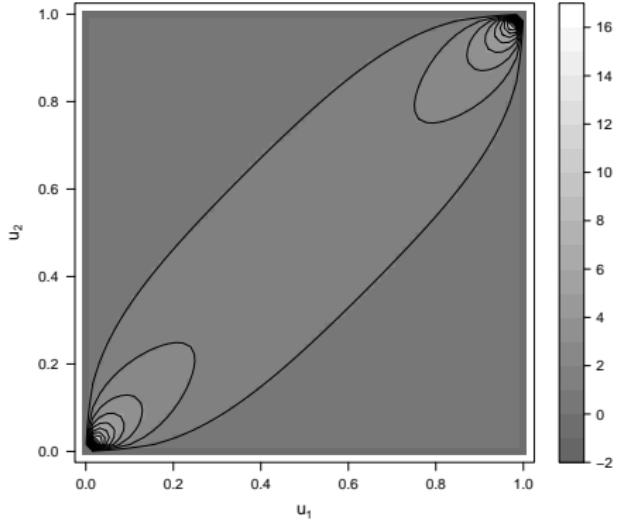


Figure 2.16 Contour plots of the densities $c_{\rho,\nu}^t$ for $\rho = 0.7$, $\nu = 3.5$ (left) and c_θ^{GH} for $\theta = 2$ (right).



2.6 Measures of association

- It is often desirable to summarize the dependence between components of a random vector by a real number, a so-called *measures of association*. They are mostly studied in the bivariate case.
- One widely known such measure is *Pearson's* (or the *linear*) *correlation coefficient* defined, for a random vector (X_1, X_2) whose components have finite variances (or, equivalently, such that $\mathbb{E}(X_1^2) < \infty$ and $\mathbb{E}(X_2^2) < \infty$), by

$$\begin{aligned}\text{Cor}(X_1, X_2) &= \frac{\text{Cov}(X_1, X_2)}{\sqrt{\text{Var}(X_1)}\sqrt{\text{Var}(X_2)}} \\ &= \frac{\mathbb{E}((X_1 - \mathbb{E}(X_1))(X_2 - \mathbb{E}(X_2)))}{\sqrt{\mathbb{E}((X_1 - \mathbb{E}(X_1))^2)}\sqrt{\mathbb{E}((X_2 - \mathbb{E}(X_2))^2)}}.\end{aligned}\quad (2.15)$$

- Some well-known properties of the linear correlation coefficient are summarized in the following proposition.

Proposition 2.6.1 (Properties of linear correlation)

Let (X_1, X_2) be a random vector whose components have finite variances. Then,

- 1) $-1 \leq \text{Cor}(X_1, X_2) \leq 1$;
- 2) $|\text{Cor}(X_1, X_2)| = 1$ if and only if there exist $a, b \in \mathbb{R}$, $a \neq 0$, such that $X_2 = aX_1 + b$ almost surely, with $a < 0$ if and only if $\text{Cor}(X_1, X_2) = -1$ and $a > 0$ if and only if $\text{Cor}(X_1, X_2) = 1$. In this case, X_1 and X_2 are perfectly linearly dependent.
- 3) If X_1 and X_2 are independent, then $\text{Cor}(X_1, X_2) = 0$.
- 4) For any $a_1 > 0$, $a_2 > 0$, or any $a_1 < 0$, $a_2 < 0$, and any $b_1, b_2 \in \mathbb{R}$,

$$\text{Cor}(a_1 X_1 + b_1, a_2 X_2 + b_2) = \text{Cor}(X_1, X_2).$$

In particular, Pearson's correlation coefficient is invariant under strictly increasing linear transformations.



- Pearson's correlation coefficient is merely a measure of linear dependence.
- Its usefulness for quantifying dependence is truly meaningful only for so-called elliptical distributions (see Section 3.1) such as the multivariate normal or t distributions; see also the fallacies in Section 2.6.1.
- We also present two rank correlation coefficients and the so-called tail-dependence coefficients.
- We focus on continuous random vectors, which implies that these measures of association depend only on the underlying unique copula and, hence, are of much broader applicability than the linear correlation coefficient.

2.6.1 Fallacies related to the correlation coefficient

We now consider fallacies related to the use of the linear correlation coefficient as a measure of dependence; see Embrechts et al. (2002).

The first fallacy can be avoided by carefully examining (2.15).

Fallacy 1 (Existence)

$\text{Cor}(X_1, X_2)$ exists for every random vector (X_1, X_2) . □

- A well-known consequence of the Cauchy–Schwarz inequality is that $|\text{Cov}(X_1, X_2)| \leq \sqrt{\text{Var}(X_1) \text{Var}(X_2)}$. Hence, both the numerator and the denominator in (2.15) exist if the second moments of X_1 and X_2 are finite. If one of these moments is infinite, $\text{Cor}(X_1, X_2)$ is not defined.
- As a counterexample to Fallacy 1, consider $X_1, X_2 \stackrel{\text{ind.}}{\sim} F(x) = 1 - x^{-3}$, $x \in [1, \infty)$. By independence (see Proposition 2.6.1), $\text{Cor}(X_1, X_2) = 0$ but $\text{Cor}(X_1^3, X_2)$ does not exist since neither $\mathbb{E}((X_1^3)^2)$ nor $\mathbb{E}(X_1^3)$ are finite.
- This counterexample can also serve as a counterexample to the following fallacy.

Fallacy 2 (Invariance)

$\text{Cor}(X_1, X_2)$ is invariant under strictly increasing transformations on $\text{ran } X_1$ or $\text{ran } X_2$. \square

The following proposition provides a useful representation of the correlation coefficient; see McNeil et al. (2005, pp. 203).

Proposition 2.6.2 (Höffding's formula)

Let (X_1, X_2) be a random vector with df H with margins F_1, F_2 and whose components have finite variances. Then

$$\text{Cor}(X_1, X_2) = \frac{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (H(x_1, x_2) - F_1(x_1)F_2(x_2)) dx_1 dx_2}{\sqrt{\text{Var}(X_1)} \sqrt{\text{Var}(X_2)}}. \quad \square$$

By Sklar's Theorem, it follows that

$$\text{Cor}(X_1, X_2) = \frac{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (C(F_1(x_1), F_2(x_2)) - F_1(x_1)F_2(x_2)) dx_1 dx_2}{\sqrt{\text{Var}(X_1)} \sqrt{\text{Var}(X_2)}},$$

where C is the copula of H . This representation leads to a counterexample to the following two fallacies and can also be used to refute Fallacy 5.

Fallacy 3 (Uniqueness)

The marginal distributions and the correlation coefficient uniquely determine the joint distribution. \square

Fallacy 4 (Uncorrelatedness implies independence)

$\text{Cor}(X_1, X_2) = 0$ implies that X_1 and X_2 are independent. \square

Example 2.6.3 (Counterexample to Fallacies 3 and 4)

By checking groundedness, uniform margins and the density, one can show that

$$C(u_1, u_2) = u_1 u_2 + h_1(u_1)h_2(u_2), \quad u_1, u_2 \in [0, 1], \quad (2.16)$$

is a copula for all continuously differentiable $h_1, h_2 : [0, 1] \rightarrow \mathbb{R}$ with $h_1(0) = h_2(0) = h_1(1) = h_2(1) = 0$ and $1 + h'_1(u_1)h'_2(u_2) \geq 0$ for all $u_1, u_2 \in (0, 1)$.

- Let $h_1(u) = 2u(u - 1/2)(u - 1)$ and $h_2(u) = \theta u(1 - u)$, $u \in [0, 1]$, $\theta \in [-1, 1]$ and $(U_1, U_2) \sim C$.
- Höffding's formula implies that

$$\begin{aligned}\text{Cor}(U_1, U_2) &= 12 \int_0^1 \int_0^1 (C(u_1, u_2) - u_1 u_2) du_1 du_2 \\ &= 12 \int_0^1 h_1(u_1) du_1 \int_0^1 h_2(u_2) du_2.\end{aligned}$$

- It can be checked that $\text{Cor}(U_1, U_2) = 0$ for all $\theta \in [-1, 1]$, so the margins and the correlation coefficient do not change for $\theta \in [-1, 1]$, while C does change with θ .

```

1 > ## Evaluate the density of C for h_1(u) = 2*u*(u-1/2)*(u-1),
2 > ## h_2(u) = theta*u*(1-u) and two different thetas
3 > u <- seq(0, 1, length.out = 20) # subdivision points in each dimension
4 > u12 <- expand.grid("u[1]" = u, "u[2]" = u) # build a grid
5 > dC <- function(u, th) 1 + th * (6 * u[,1] * (u[,1]-1) + 1) * (1 - 2*u[,2])
6 > wireframe2(cbind(u12, "c(u[1],u[2])" = dC(u12, th = -1)))
7 > wireframe2(cbind(u12, "c(u[1],u[2])" = dC(u12, th = 1)))

```

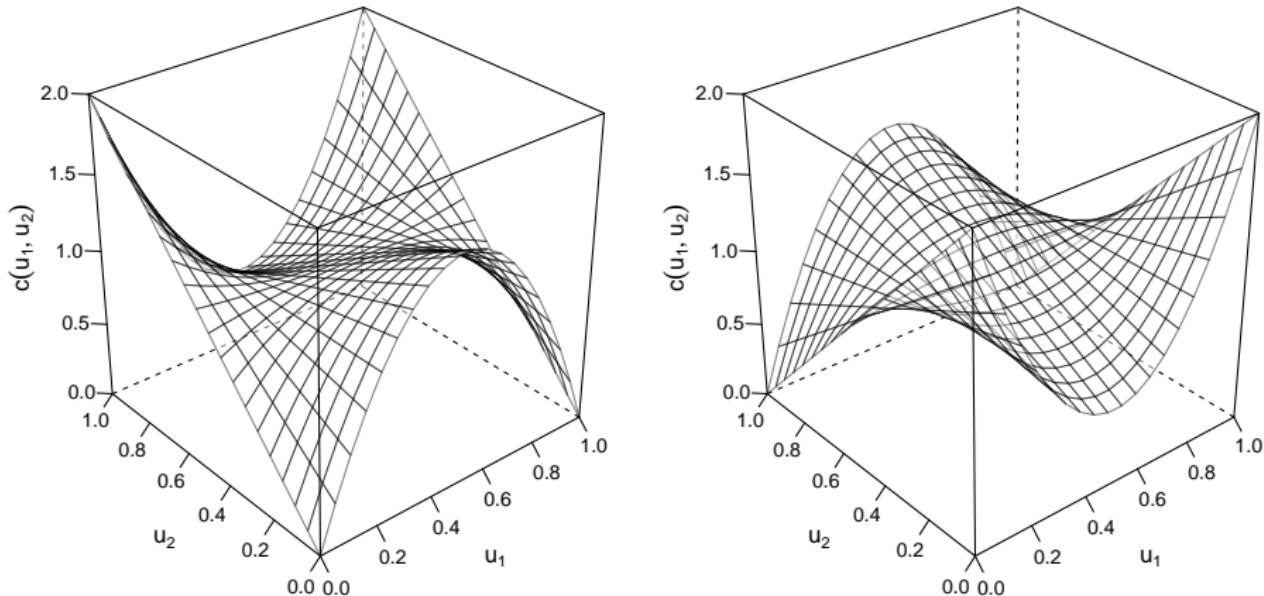


Figure 2.17 Density c of C in (2.16) for $\theta = -1$ (left) and $\theta = 1$ (right).

Since $C \neq \Pi$ for all $\theta \in [-1, 1] \setminus \{0\}$, this example also serves as a counterexample to Fallacy 4. Note that Fallacy 4 (another one would be $X_1 \sim N(0, 1)$ and $X_2 = X_1^2$). □

Example 2.6.4 (Uncorrelatedness versus independence)

- Another counterexample to Fallacies 3 and 4 can be constructed by a mixture of the two Fréchet–Hoeffding bounds W and M .
- Let $Z \sim N(0, 1)$ and define $X_1 = Z$ and $X_2 = VZ$, where V takes its values in $\{-1, 1\}$, each with probability $1/2$.
- Conditional on $V = -1$ (respectively, $V = 1$), the copula C of (X_1, X_2) is thus W (respectively, M). In other words, $C(u_1, u_2) = \lambda M(u_1, u_2) + (1 - \lambda)W(u_1, u_2)$ with $\lambda = 1/2$.
- Since $\text{Cor}(X_1, X_2) = \mathbb{E}(X_1 X_2) = \mathbb{E}(V)\mathbb{E}(Z^2) = \mathbb{E}(V) = 0$, X_1 and X_2 are uncorrelated but not independent since $C \neq \Pi$.
- Consider a plot of $n = 1000$ realizations of (X_1, X_2) and (Y_1, Y_2) (independent $Y_1, Y_2 \sim N(0, 1)$). Both have $N(0, 1)$ margins and $\rho = 0$.
- We also perform a test of uncorrelatedness on the realizations of (X_1, X_2) using `cor.test()`:

```

1 > n <- 1000
2 > set.seed(314)
3 > Z <- rnorm(n)
4 > U <- runif(n)
5 > V <- rep(1, n)
6 > V[U < 1/2] <- -1 # => V in {-1,1}, each with probability 1/2
7 > X <- cbind(Z, Z*V) # (X_1,X_2)
8 > stopifnot(cor.test(X[,1], X[,2])$p.value >= 0.05) # H0: 'cor=0' not rejected
9 > Y <- matrix(rnorm(n * 2), ncol = 2) # independent N(0,1)
10 > plot(X, xlab = quote(X[1]), ylab = quote(X[2]))
11 > plot(Y, xlab = quote(Y[1]), ylab = quote(Y[2]))

```

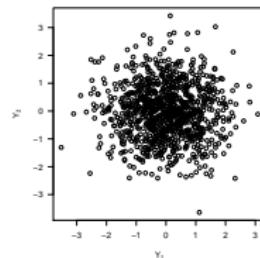
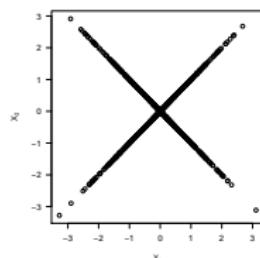


Figure 2.18 $n = 1000$ independent realizations from (X_1, X_2) (left) and (Y_1, Y_2) (right), both having $N(0, 1)$ margins and zero correlation; the copula of (X_1, X_2) (respectively, (Y_1, Y_2)) is a mixture between the Fréchet–Höffding bounds W and M (respectively, the independence copula). \square

- Let (X_1, X_2) and (X'_1, X'_2) be continuous random vectors both with the same marginal dfs F_1, F_2 but with different copulas C and C' , respectively, such that, for any $\mathbf{u} \in [0, 1]^d$, $C(\mathbf{u}) \leq C'(\mathbf{u})$. By Höffding's formula, $\text{Cor}(X_1, X_2) \leq \text{Cor}(X'_1, X'_2)$.
- The Fréchet–Höffding bounds imply that, for any continuous random vector (X_1, X_2) with marginal dfs F_1, F_2 with finite variances,

$$\underline{\text{Cor}}(X_1, X_2) \leq \text{Cor}(X_1, X_2) \leq \overline{\text{Cor}}(X_1, X_2),$$

where $\underline{\text{Cor}}(X_1, X_2)$, $\overline{\text{Cor}}(X_1, X_2)$ are obtained for W , M , respectively.

- It is a fallacy to believe that, for any margins, $\underline{\text{Cor}}(X_1, X_2) = -1$ and that $\overline{\text{Cor}}(X_1, X_2) = 1$.

Fallacy 5 (Attainable correlations)

Given margins F_1, F_2 , all $\text{Cor}(X_1, X_2) \in [-1, 1]$ can be attained by choosing a suitable copula for (X_1, X_2) . □

Example 2.6.5 (Counterexample to Fallacy 5)

- A classical counterexample for Fallacy 5 consists of random vectors (X_1, X_2) with lognormally distributed margins, that is, $X_j \sim \text{LN}(0, \sigma_j^2)$ or, equivalently, $X_j \stackrel{d}{=} \exp(\sigma_j Z_j)$ for $Z_j \sim N(0, 1)$, $j \in \{1, 2\}$.
- Minimal and maximal correlations for the Fréchet class with these marginals can be computed as

$$\underline{\text{Cor}}(X_1, X_2) = \rho_{\sigma_1, -\sigma_2} \quad \text{and} \quad \overline{\text{Cor}}(X_1, X_2) = \rho_{\sigma_1, \sigma_2},$$

where $\rho_{x,y} = \frac{\exp((x+y)^2/2) - \exp((x^2+y^2)/2)}{\sqrt{(\exp(x^2)-1)\exp(y^2)}} \sqrt{(\exp(x^2)-1)\exp(y^2)}$.

```
1 > ## Function to compute the correlation bounds for LN(0, sigma_.^2) margins
2 > corBoundLN <- function(s, bound = c("max", "min"))
3 {
4     ## s = (sigma_1, sigma_2)
5     if(!is.matrix(s)) s <- rbind(s)
6     bound <- match.arg(bound)
7     if(bound == "min") s[,2] <- -s[,2]
8     (exp((s[,1]+s[,2])^2/2)-exp((s[,1]^2+s[,2]^2)/2)) /
9         sqrt(expm1(s[,1]^2)*exp(s[,1]^2)*expm1(s[,2]^2)*exp(s[,2]^2))
10 }
```

```

11 > ## Evaluate correlation bounds on a grid
12 > s <- seq(0.01, 5, length.out = 20) # subdivision points in each dimension
13 > s12 <- expand.grid("sigma[1]" = s, "sigma[2]" = s) # build a grid
14 > ## Plots
15 > wireframe2(cbind(s12, 'underline(Cor)(sigma[1],sigma[2])' =
16 >                           corBoundLN(s12, bound = "min")))
17 > wireframe2(cbind(s12, 'bar(Cor)(sigma[1],sigma[2])' = corBoundLN(s12)))

```

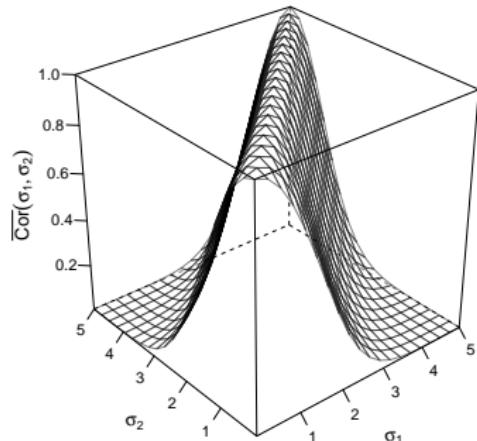
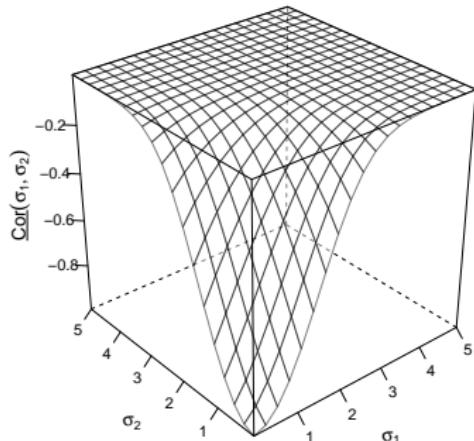


Figure 2.19 Minimal (left) and maximal (right) attainable correlations under $\text{LN}(0, \sigma_1^2)$ and $\text{LN}(0, \sigma_2^2)$ margins.

To conclude, the main limitations of the linear correlation coefficient $\text{Cor}(X_1, X_2)$ as a measure of association are:

- 1) $\text{Cor}(X_1, X_2)$ does not exist for all random vectors (X_1, X_2) (only for those with finite second moments);
- 2) $\text{Cor}(X_1, X_2)$ depends on the marginal dfs of (X_1, X_2) even when the latter are continuous (it can thus not be expressed in terms of the unique underlying copula alone);
- 3) $\text{Cor}(X_1, X_2)$ is invariant only under strictly increasing linear transformations (not under strictly increasing transformations in general).

2.6.2 Rank correlation measures

By only depending on the underlying copula C , rank correlation coefficients overcome several of the aforementioned limitations of the linear correlation coefficient. We focus on two examples: Spearman's rho and Kendall's tau.

Definition 2.6.6 (Spearman's rho, Kendall's tau)

Let (X_1, X_2) be a bivariate random vector with continuous marginal dfs F_1 and F_2 .

- 1) The *(population version of) Spearman's rho* is defined by

$$\rho_s = \rho_s(X_1, X_2) = \text{Cor}(F_1(X_1), F_2(X_2)).$$

- 2) Let (X'_1, X'_2) be an independent copy of (X_1, X_2) . The *(population version of) Kendall's tau* is defined as

$$\tau = \tau(X_1, X_2) = \mathbb{E}(\text{sign}((X_1 - X'_1)(X_2 - X'_2))),$$

where $\text{sign}(x)$ denotes the *sign* of x , that is,

$$\text{sign}(x) = \begin{cases} -1, & \text{if } x < 0, \\ 0, & \text{if } x = 0, \\ 1, & \text{if } x > 0. \end{cases}$$

□

- Spearman's rho is simply the linear correlation coefficient of the random vector $(F_1(X_1), F_2(X_2))$ obtained from (X_1, X_2) by marginally applying the probability transformation. It (thus) always exists.
- By Lemma 2.4.1, Spearman's rho only depends on the underlying C , and thus not on the marginal dfs F_1 and F_2 . Hence, ρ_s measures dependence independently of the margins.
- These immediate results already show that Spearman's rho is not affected by Fallacies 1 and 2. One can also show that Spearman's rho and Kendall's tau are measures of concordance; see Scarsini (1984) for an axiomatic definition.
- To define concordance (and to better understand the definition of Kendall's tau), consider two points in \mathbb{R}^2 denoted by (x_1, x_2) and (x'_1, x'_2) . These points are said to be concordant if $(x_1 - x'_1)(x_2 - x'_2) > 0$ (so if the slope of the line through the two points is positive) and to be discordant if $(x_1 - x'_1)(x_2 - x'_2) < 0$.

- Kendall's tau can be written as

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

that is, Kendall's tau is the probability of concordance minus the probability of discordance of the random pairs (X_1, X_2) and (X'_1, X'_2) .

- By the assumption of continuous margins, one obtains that

$$\tau = 2\mathbb{P}((X_1 - X'_1)(X_2 - X'_2) > 0) - 1. \quad (2.17)$$

- Like Spearman's rho, Kendall's tau can be expressed in terms of the linear correlation coefficient. This can be seen from (2.17) via

$$\begin{aligned}\tau &= 4\mathbb{P}(X_1 \leq X'_1, X_2 \leq X'_2) - 1 = \frac{\mathbb{E}(1(X_1 \leq X'_1, X_2 \leq X'_2)) - \frac{1}{2} \cdot \frac{1}{2}}{\sqrt{\frac{1}{2}\left(1 - \frac{1}{2}\right)\frac{1}{2}\left(1 - \frac{1}{2}\right)}} \\ &= \frac{\mathbb{E}(1(X_1 \leq X'_1)1(X_2 \leq X'_2)) - \mathbb{E}(1(X_1 \leq X'_1))\mathbb{E}(1(X_2 \leq X'_2))}{\sqrt{\text{Var}(1(X_1 \leq X'_1)) \text{Var}(1(X_2 \leq X'_2))}} \\ &= \text{Cor}(1(X_1 \leq X'_1), 1(X_2 \leq X'_2)).\end{aligned}$$

Proposition 2.6.7 (Representation of Spearman's rho and Kendall's tau)

Let (X_1, X_2) be a bivariate random vector with continuous marginal dfs and copula C . Then,

$$\rho_s = \rho_s(C) = 12 \int_{[0,1]^2} C(\mathbf{u}) d\mathbf{u} - 3 = 12 \int_{[0,1]^2} u_1 u_2 dC(\mathbf{u}) - 3 \quad (2.18)$$

and

$$\tau = \tau(C) = 4 \int_{[0,1]^2} C(\mathbf{u}) dC(\mathbf{u}) - 1. \quad (2.19)$$

□

- Spearman's rho and Kendall's tau depend only on the underlying copula.
- This proposition can also be used to show that both measures of association satisfy the defining axioms of a measure of concordance.

- As such they share the property that $\rho_s(W) = \tau(W) = -1$, $\rho_s(M) = \tau(M) = 1$. This shows that an analog of Fallacy 5 does not apply to rank correlations measures.
- For ρ_s and τ in particular, one can also show the converse: If one of these measures is -1 or 1 , then the copula C of (X_1, X_2) must be W or M , respectively.
- It also follows from the axioms of a measure of concordance that if $C = \Pi$, then the measure is 0 . Similarly to Fallacy 4, however, the converse is not true in general which can be shown, for instance, by considering the uniform distribution on the unit circle; see Embrechts et al. (2002, Proposition 3).
- Consider $\{C_\theta : \theta \in \Theta\}$, where $\Theta \subseteq \mathbb{R}$. For many such copula families,

$$g_{\rho_s}(\theta) = \rho_s(C_\theta) \quad \text{and} \quad g_\tau(\theta) = \tau(C_\theta), \quad \theta \in \Theta,$$

are one-to-one. For example:

- ▶ Clayton: $g_\tau(\theta) = \theta / (\theta + 2)$, $\theta \in (0, \infty)$;
- ▶ Gumbel–Hougaard: $g_\tau(\theta) = 1 - 1/\theta$, $\theta \in [1, \infty)$;
- ▶ Normal: $g_{\rho_s}(\theta) = (6/\pi) \arcsin(\theta/2)$ and $g_\tau(\theta) = (2/\pi) \arcsin \theta$, $\theta \in [-1, 1]$.
- By considering $g_{\rho_s}^{-1}$ and g_τ^{-1} , one can thus obtain θ . This idea lies at the core of the well-known method-of-moments estimators of copula parameters; see Section 4.1.2 for more details.

Example 2.6.8 (`rho()`, `iRho()`, `tau()` and `iTau()`)

In the R package `copula`, ρ_s and τ are called `rho()` and `tau()`, respectively. They can be used to evaluate g_{ρ_s} and g_τ :

```

1 > theta <- -0.7
2 > stopifnot(all.equal(rho(normalCopula(theta)), 6 / pi * asin(theta / 2)))
3 > stopifnot(all.equal(tau(normalCopula(theta)), 2 / pi * asin(theta)))
4 > theta <- 2
5 > stopifnot(all.equal(tau(claytonCopula(theta)), theta / (theta + 2)))
6 > stopifnot(all.equal(tau(gumbelCopula(theta)), 1 - 1 / theta))

```

$g_{\rho_s}^{-1}$, g_{τ}^{-1} are called `iRho()`, `iTau()` and are “vectorized”:

```
1 > theta <- (0:8)/16
2 > stopifnot(all.equal(iRho(normalCopula(), rho = 6/pi * asin(theta/2)), theta))
3 > stopifnot(all.equal(iTau(normalCopula(), tau = 2/pi * asin(theta)), theta))
4 > theta <- 1:20
5 > stopifnot(all.equal(iTau(claytonCopula(), theta / (theta + 2)), theta))
6 > stopifnot(all.equal(iTau(gumbelCopula(), 1 - 1 / theta), theta))
```

g_{ρ_s} and g_{τ} do not necessarily admit an analytical expression. In that case, a numerical approximation is used; see Kojadinovic and Yan (2010a, Appendix A):

```
1 > theta <- 3
2 > iRho(claytonCopula(), rho = rho(claytonCopula(theta)))
```

```
[1] 3.000007
```



Given a random sample $(X_{11}, X_{12}), \dots, (X_{n1}, X_{n2})$ from (X_1, X_2) , how could Spearman's rho or Kendall's tau be estimated?

- Consider Spearman's rho. As being the linear correlation coefficient of $(F_1(X_1), F_2(X_2))$, ρ_s can be estimated by
 - 1) estimating the unknown marginal dfs F_1 and F_2 ; then
 - 2) estimating the unobservable random sample $(F_1(X_{11}), F_2(X_{12})), \dots, (F_1(X_{n1}), F_2(X_{n2}))$; and then
 - 3) computing the correlation coefficient from the "pseudo-observations" of $(F_1(X_1), F_2(X_2))$.
- Estimating F_1 and F_2 empirically amounts to working with the sample of bivariate ranks $(R_{11}, R_{12}), \dots, (R_{n1}, R_{n2})$, where R_{ij} is the rank of X_{ij} among X_{1j}, \dots, X_{nj} , $j \in \{1, 2\}$.
- A natural estimator of ρ_s (referred to as the *sample version of Spearman's rho*) is then simply

$$\rho_{s,n} = \frac{\sum_{i=1}^n (R_{i1} - \bar{R}_1)(R_{i2} - \bar{R}_2)}{\sqrt{\sum_{i=1}^n (R_{i1} - \bar{R}_1)^2} \sqrt{\sum_{i=1}^n (R_{i2} - \bar{R}_2)^2}}, \quad (2.20)$$

where $\bar{R}_1 = \bar{R}_2 = (n + 1)/2$ is the mean rank of the two component series. In other words, $\rho_{s,n}$ is merely the sample linear correlation coefficient computed from the sample of bivariate ranks.

- This estimator can also be seen as a “plug-in estimator”: if one starts from (2.18) and replaces C by a non-parametric estimator of the latter, see Section 4.2.1, one recovers (2.20).

Let us now turn to the estimation of Kendall's tau.

- Starting from a bivariate random sample $(X_{11}, X_{12}), \dots, (X_{n1}, X_{n2})$ from (X_1, X_2) and replacing the expectation operator in Definition 2.6.6 by a sample mean, a natural estimator of Kendall's tau (referred to as the *sample version of Kendall's tau*) is given by

$$\tau_n = \frac{1}{\binom{n}{2}} \sum_{1 \leq i < j \leq n} \text{sign}((X_{i1} - X_{j1})(X_{i2} - X_{j2})).$$

- The latter can be rewritten as a sample version of (2.17):

$$\tau_n = 2 \frac{p_n}{\binom{n}{2}} - 1 = \frac{4p_n}{n(n-1)} - 1, \quad (2.21)$$

where p_n is the number of concordant pairs in the sample.

- Similar to Spearman's rho, τ_n can be seen as resulting from plugging-in a non-parametric estimator of C (see Section 4.2.1) into (2.19).

Example 2.6.9 (Estimating Spearman's rho and Kendall's tau)

Spearman's rho can be estimated with `cor(, method = "spearman")`:

```

1 > theta <- iRho(claytonCopula(), rho = 0.6) # true Spearmans' rho = 0.6
2 > set.seed(974)
3 > U <- rCopula(1000, copula = claytonCopula(theta))
4 > rho.def <- cor(apply(U, 2, rank)) [1,2]      # Spearman's rho manually
5 > rho.R   <- cor(U, method = "spearman") [1,2] # Spearman's rho from R
6 > stopifnot(all.equal(rho.def, rho.R)) # the same
7 > rho.R # indeed close to 0.6

```

```
[1] 0.5915302
```

Kendall's tau can be estimated with `cor(, method = "kendall")`:

```
1 > theta <- iTau(normalCopula(), tau = -0.5) # true Kendall's tau = -0.5
2 > set.seed(974)
3 > U <- rCopula(1000, copula = normalCopula(theta))
4 > p.n <- 0
5 > for(i in 1:(n-1)) # number of concordant pairs (obviously inefficient)
6     for(j in (i+1):n)
7         if(prod(apply(U[c(i,j),], 2, diff)) > 0) p.n <- p.n + 1
8 > tau.def <- 4 * p.n / (n * (n - 1)) - 1    # Kendall's tau manually
9 > tau.R <- cor(U, method = "kendall")[1,2] # Kendall's tau from R
10 > stopifnot(all.equal(tau.def, tau.R)) # the same
11 > tau.R # close to -0.5
```

```
[1] -0.5308388
```



Example 2.6.10 (Spearman's rho and Kendall's tau under counter-and comonotonicity)

We can also verify that $\rho_{s,n}$ and τ_n are -1 under countermonotonicity and 1 under comonotonicity:

```
1 > set.seed(75)
2 > X <- rnorm(100)
3 > Y <- -X^3 # perfect negative dependence
4 > rho.counter <- cor(X, Y, method = "spearman")
5 > tau.counter <- cor(X, Y, method = "kendall")
6 > stopifnot(rho.counter == -1, tau.counter == -1)
7 > Z <- exp(X) # perfect positive dependence
8 > rho.co <- cor(X, Z, method = "spearman")
9 > tau.co <- cor(X, Z, method = "kendall")
10 > stopifnot(rho.co == 1, tau.co == 1)
```



To conclude, Spearman's rho and Kendall's tau

- 1) always exist (and are not limited to continuous random vectors with finite second moments; compare with Fallacy 1);
- 2) are invariant under strictly increasing transformations (compare with Fallacy 2); and
- 3) can reach any value in $[-1, 1]$ (compare with Fallacy 5).

- Analogs of Fallacies 3 and 4 still apply to rank correlation coefficients. This cannot be avoided since we know by Sklar's Theorem that a function on $[0, 1]^d$ completely determines the dependence between the components of a continuous random vector, not a single number.
- Nevertheless, rank correlation measures are useful. For instance, for bivariate data sets that exhibit monotone trends, sample versions of Spearman's rho and Kendall's tau provide an idea of the "average strength of association".
- One advantage of Kendall's tau is that there are (semi-)analytical formulas known for two popular classes of copulas, elliptical and Archimedean copulas; see Sections 3.1 and 3.2.

2.6.3 Tail dependence coefficients

- Coefficients of tail dependence aim at summarizing the dependence in the (joint) tails of bivariate distributions.
- It is known that the tails of a bivariate t distribution with low degrees of freedom and correlation $|\rho| < 1$ have more probability mass than the tails of a normal distribution with correlation ρ .
- This (and many other tail-related applications of copula theory) can be quantified with the notion of tail dependence.

Example 2.6.11 (Four distributions with $N(0, 1)$ margins and a Kendall's tau of 0.7)

Consider the Fréchet class with standard normal margins, and let us investigate how the choice of the copula affects the dependence in the tails. Consider four one-parameter copulas with a Kendall's tau of 0.7 and generate random samples of size $n = 10\,000$:

```

1 > ## Kendall's tau and corresponding copula parameters
2 > tau <- 0.7
3 > theta.n <- iTau(normalCopula(), tau = tau)
4 > theta.t <- iTau(tCopula(df = 3), tau = tau)
5 > theta.c <- iTau(claytonCopula(), tau = tau)
6 > theta.g <- iTau(gumbelCopula(), tau = tau)
7 > ## Samples from the corresponding 'mvdc' objects
8 > set.seed(271)
9 > n <- 10000
10 > N01.marg <- list(list(mean = 0, sd = 1), list(mean = 0, sd = 1))
11 > ## For the normal copula
12 > h.n <- mvdc(normalCopula(theta.n), c("norm", "norm"), N01.marg)
13 > X.n <- rMvdc(n, mvdc = h.n)
14 > ## For the t copula
15 > h.t <- mvdc(tCopula(theta.t, df = 3), c("norm", "norm"), N01.marg)
16 > X.t <- rMvdc(n, mvdc = h.t)
17 > ## For the Clayton copula
18 > h.c <- mvdc(claytonCopula(theta.c), c("norm", "norm"), N01.marg)
19 > X.c <- rMvdc(n, mvdc = h.c)
20 > ## For the Gumbel-Hougaard copula
21 > h.g <- mvdc(gumbelCopula(theta.g), c("norm", "norm"), N01.marg)
22 > X.g <- rMvdc(n, mvdc = h.g)
23 > #' @title Function for producing one scatter plot
24 > #' @param X data
25 > #' @param qu (lower and upper) quantiles to consider
26 > #' @param lim (x- and y-axis) limits

```

```

27 > #' @param ... additional arguments passed to the underlying plot functions
28 > #' @return invisible()
29 > plotCorners <- function(X, qu, lim, smooth = FALSE, ...)
30 {
31   plot(X, xlim = lim, ylim = lim, xlab = quote(X[1]), ylab = quote(X[2])),
32   col = adjustcolor("black", 0.5), ...) # or pch = 16
33   abline(h = qu, v = qu, lty = 2, col = adjustcolor("black", 0.6))
34   ll <- sum(apply(X <= qu[1], 1, all)) * 100 / n
35   ur <- sum(apply(X >= qu[2], 1, all)) * 100 / n
36   mtext(sprintf("Lower left: %.2f%%, upper right: %.2f%%", ll, ur),
37         cex = 0.9, side = 1, line = -1.5)
38   invisible()
39 }
40 > ## Plots
41 > a. <- 0.005
42 > q <- qnorm(c(a., 1 - a.)) # a- and (1-a)-quantiles of N(0,1)
43 > lim <- range(q, X.n, X.t, X.c, X.g)
44 > lim <- c(floor(lim[1]), ceiling(lim[2]))
45 > plotCorners(X.n, qu = q, lim = lim, cex = 0.4)
46 > plotCorners(X.t, qu = q, lim = lim, cex = 0.4)
47 > plotCorners(X.c, qu = q, lim = lim, cex = 0.4)
48 > plotCorners(X.g, qu = q, lim = lim, cex = 0.4)

```

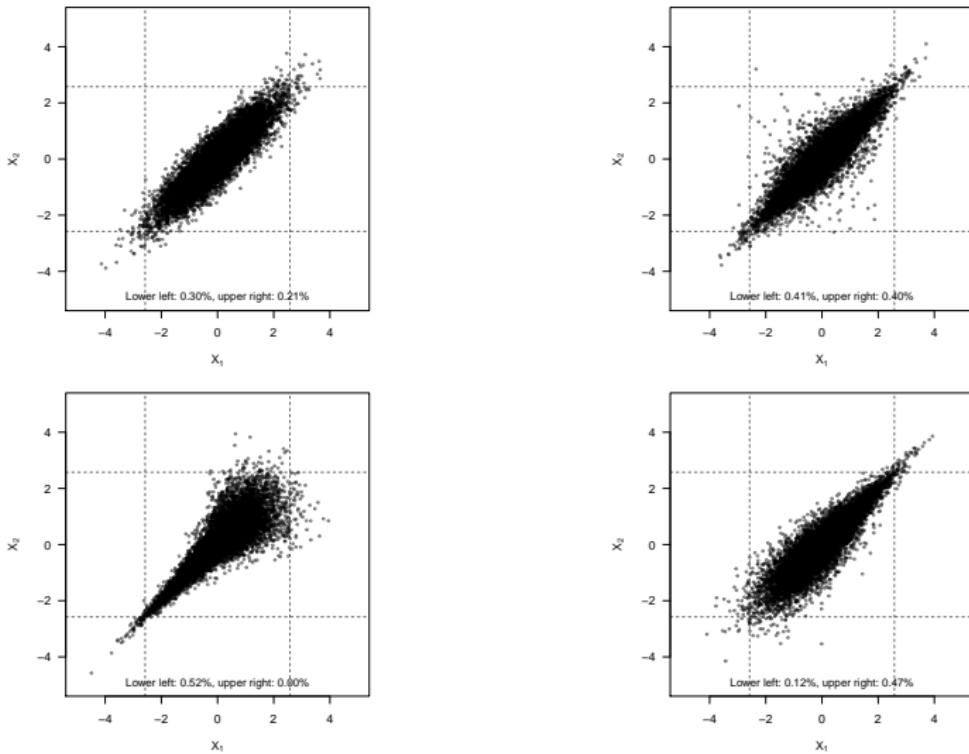


Figure 2.20 Scatter plots of four samples of size $n = 10000$ from bivariate distributions with $N(0, 1)$ margins and different copulas (top left: normal; top right: t_3 ; bottom left: Clayton; bottom right: Gumbel–Hougaard), all with a Kendall's tau of 0.7. The dashed lines indicate the $N(0, 1)$ 0.005- and 0.995-quantiles.

- The four distributions have different behavior in their bivariate tails although they all share the same margins and Kendall's tau.
- The differences in the bivariate tails are solely a consequence of the underlying copulas and, in particular, of the notion of tail dependence.
- For the normal copula (top left), there seems to be no dependence in the lower left and upper right corners, while the t copula with 3 degrees of freedom (top right) appears to have more mass and more structure in the lower and upper tail (thus hinting at “tail dependence”).
- For the Clayton copula (bottom left), small values of one variable frequently seem to be accompanied by small values of the other variable (thus hinting at “lower tail dependence”), whereas large values of one variable do not necessarily seem to imply large values of the other variable (thus hinting at “upper tail independence”).
- For the Gumbel–Hougaard copula (bottom right), this behavior seems to be reversed. □

Definition 2.6.12 (Coefficients of tail dependence)

Let (X_1, X_2) be a random vector with marginal dfs F_1, F_2 . Provided that the limits exists, the *coefficient of lower* and *upper tail dependence* of X_1 and X_2 are defined by

$$\lambda_l = \lambda_l(X_1, X_2) = \lim_{q \downarrow 0} \mathbb{P}(X_2 \leq F_2^\leftarrow(q) \mid X_1 \leq F_1^\leftarrow(q)),$$

$$\lambda_u = \lambda_u(X_1, X_2) = \lim_{q \uparrow 1} \mathbb{P}(X_2 > F_2^\leftarrow(q) \mid X_1 > F_1^\leftarrow(q)),$$

respectively. If $\lambda_l(X_1, X_2) \in (0, 1]$ (respectively, $\lambda_u(X_1, X_2) \in (0, 1]$), then X_1 and X_2 are said to be *lower* (respectively, *upper*) *tail dependent*. If $\lambda_l(X_1, X_2) = 0$ (respectively, $\lambda_u(X_1, X_2) = 0$), X_1 and X_2 are *lower* (respectively, *upper*) *tail independent*. □

Like rank correlations, when the marginal dfs F_1 and F_2 are continuous, coefficients of tail dependence are measures of dependence that depend only on the underlying unique copula.

Proposition 2.6.13 (Representation of the coefficients of tail dependence)

Let (X_1, X_2) be a bivariate random vector with continuous margins and copula C . Then,

$$\lambda_l = \lambda_l(C) = \lim_{q \downarrow 0} \frac{C(q, q)}{q},$$

$$\lambda_u = \lambda_u(C) = \lim_{q \uparrow 1} \frac{1 - 2q + C(q, q)}{1 - q} = \lim_{q \downarrow 0} \frac{\bar{C}(q, q)}{q}. \quad \square$$

- Calculating λ_l and λ_u is often straightforward: typically by using the above formulas and l'Hôpital's rule.
- For example, for Archimedean copulas with differentiable generator ψ ,

$$\lambda_l = \lim_{t \rightarrow \infty} \frac{\psi(2t)}{\psi(t)} = 2 \lim_{t \rightarrow \infty} \frac{\psi'(2t)}{\psi'(t)}, \quad (2.22)$$

$$\lambda_u = 2 - \lim_{t \downarrow 0} \frac{1 - \psi(2t)}{1 - \psi(t)} = 2 - 2 \lim_{t \downarrow 0} \frac{\psi'(2t)}{\psi'(t)}. \quad (2.23)$$

- For a **Clayton copula** with parameter $\theta \in (0, \infty)$, we obtain

$$\lambda_l = 2^{-1/\theta} \quad \text{and} \quad \lambda_u = 0,$$

When modeling positive association, bivariate Clayton copulas are thus **lower tail dependent but upper tail independent**; note that $\lambda_l \rightarrow 1$ as $\theta \rightarrow \infty$ since Clayton copulas tend to M as $\theta \rightarrow \infty$.

- For a **Gumbel–Hougaard copula** with parameter θ , one has

$$\lambda_l = 0 \quad \text{and} \quad \lambda_u = 2 - 2^{1/\theta}, \quad \theta \in [1, \infty).$$

A Gumbel–Hougaard copula is thus **upper tail dependent** unless it is the independence copula, and always **lower tail independent**; note that $\lambda_u \rightarrow 1$ as $\theta \rightarrow \infty$ since Gumbel–Hougaard copulas tend to M as $\theta \rightarrow \infty$.

- Proposition 2.6.13 implies that $\lambda_l = \lambda_u$ for radially symmetric copulas.
Using results about conditional normal and t distributions, one can obtain $\lambda = \lambda_l = \lambda_u$ for normal and t copulas.
- ▶ For the bivariate normal copula C_ρ^n ,

$$\lambda = 1(\rho = 1).$$

Normal copulas are thus tail independent unless they equal M .

- ▶ For the bivariate t copula $C_{\rho,\nu}^t$,

$$\lambda = 2t_{\nu+1} \left(-\sqrt{\frac{(\nu+1)(1-\rho)}{1+\rho}} \right),$$

where $t_{\nu+1}$ denotes the df of the Student t distribution with $\nu + 1$ degrees of freedom.

Example 2.6.14 (Computing the coefficients of tail dependence)

The coefficients of tail dependence can be computed with `lambda()`:

```
1 > ## Clayton copula
2 > theta <- 3
3 > lam.c <- lambda(claytonCopula(theta))
4 > stopifnot(all.equal(lam.c[["lower"]], 2^(-1/theta)),
5           all.equal(lam.c[["upper"]], 0))
6 > ## Gumbel-Hougaard copula
7 > lam.g <- lambda(gumbelCopula(theta))
8 > stopifnot(all.equal(lam.g[["lower"]], 0),
9           all.equal(lam.g[["upper"]], 2-2^(1/theta)))
10 > ## Normal copula
11 > rho <- 0.7
12 > nu <- 3
13 > lam.n <- lambda(normalCopula(rho))
14 > stopifnot(all.equal(lam.n[["lower"]], 0),
15           all.equal(lam.n[["lower"]], lam.n[["upper"]]))
16 > ## t copula
17 > lam.t <- lambda(tCopula(rho, df = nu))
18 > stopifnot(all.equal(lam.t[["lower"]]),
19           2*pt(-sqrt((nu+1)*(1-rho)/(1+rho)), df = nu + 1),
20           all.equal(lam.t[["lower"]], lam.t[["upper"]]))
```



Example 2.6.15 (Tail dependence of t copulas)

Graphs of λ for $C_{\rho,\nu}^t$ as a function of ρ (left) and ν (right):

```
1 > ## Coefficient of tail dependence as a function of rho
2 > rho <- seq(-1, 1, by = 0.01)
3 > nu <- c(3, 4, 8, Inf)
4 > n.nu <- length(nu)
5 > lam.rho <- sapply(nu, function(nu.) # (rho, nu) matrix
6   sapply(rho, function(rho.) lambda(tCopula(rho., df = nu.))[[["lower"]]]))
7 > expr.rho <- as.expression(lapply(1:n.nu, function(j)
8   bquote(nu == .(if(nu[j] == Inf) quote(infinity) else nu[j]))))
9 > matplot(rho, lam.rho, type = "l", lty = 1, lwd = 2, col = 1:n.nu,
10   xlab = quote(rho), ylab = quote(lambda))
11 > legend("topleft", legend = expr.rho, bty = "n", lwd = 2, col = 1:n.nu)
12 > ## Coefficient of tail dependence as a function of nu
13 > nu. <- c(seq(3, 12, by = 0.2), Inf)
14 > rho. <- c(-1, -0.5, 0, 0.5, 1)
15 > n.rho <- length(rho.)
16 > lam.nu <- sapply(rho., function(rh) # (nu, rho) matrix
17   sapply(nu., function(nu) lambda(tCopula(rh, df = nu))[[["lower"]]]))
18 > expr <- as.expression(lapply(1:n.rho, function(j) bquote(rho == .(rho.[j]))))
19 > matplot(nu., lam.nu, type = "l", lty = 1, lwd = 2, col = 1:n.rho,
20   xlab = quote(nu), ylab = quote(lambda))
21 > legend("right", expr, bty = "n", lwd = 2, col = 1:n.rho)
```

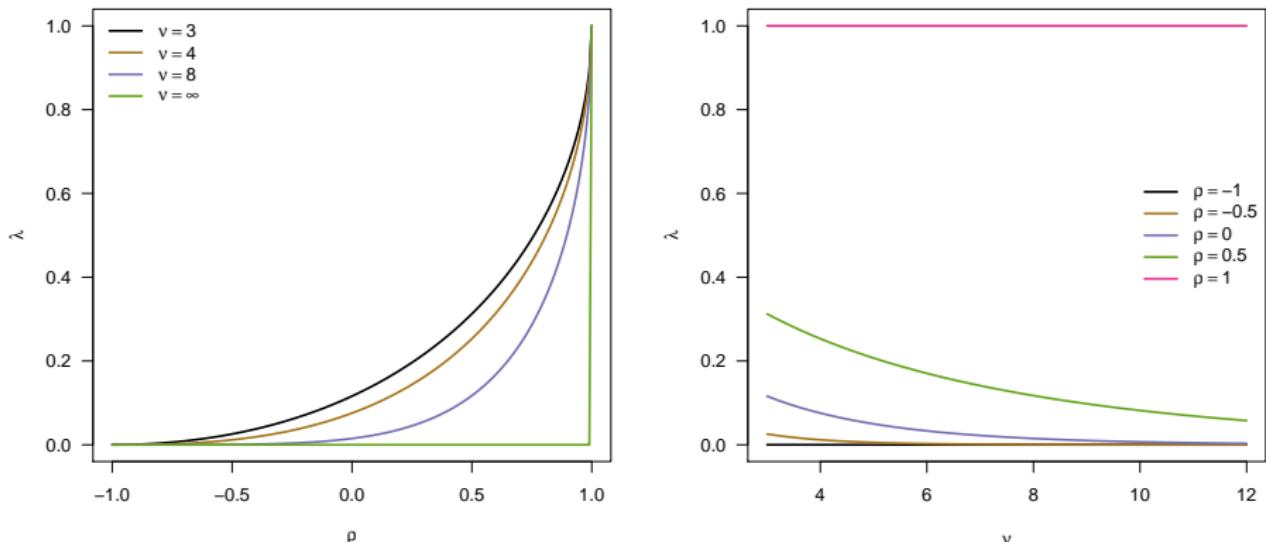


Figure 2.21 Coefficient of tail dependence $\lambda (= \lambda_l = \lambda_u)$ of $C_{\rho,\nu}^t$ as a function of the correlation parameter ρ (left) and the degrees of freedom ν (right).

For fixed, finite ν , λ increases as ρ increases; for fixed $|\rho| < 1$, λ increases as ν decreases. \square

Example 2.6.16 (Effect of ρ and ν on $u \mapsto \mathbb{P}(U_1 > u, U_2 > u)$ for t copulas)

We now illustrate how ρ and ν affect the (joint) tail probability function $u \mapsto \mathbb{P}(U_1 > u, U_2 > u) = \mathbb{P}(1 - U_1 > u, 1 - U_2 > u) = \mathbb{P}(U_1 \leq 1 - u, U_2 \leq 1 - u) = C_{\rho, \nu}^t(1 - u, 1 - u)$.

```
1 > ## Note: All calculations here are deterministic
2 > u <- seq(0.95, to = 0.9999, length.out = 128) # levels u of P(U_1 > u, U_2 > u)
3 > rho <- c(0.75, 0.5) # correlation parameter rho
4 > nu <- c(3, 4, 8, Inf) # degrees of freedom
5 > len <- length(rho) * length(nu)
6 > tail.prob <- matrix(u, nrow = length(u), ncol = 1 + len) # tail probabilities
7 > expr <- vector("expression", length = len) # vector of expressions
8 > lty <- cols <- numeric(len) # line types and colors
9 > for(i in seq_along(rho)) { # rho
10   for(j in seq_along(nu)) { # degrees of freedom
11     k <- length(nu) * (i - 1) + j
12     ## Create the copula
13     cop <- ellipCopula("t", param = rho[i], df = nu[j])
14     ## Evaluate P(U_1 > u, U_2 > u) = P(U_1 <= 1 - u, U_2 <= 1 - u)
15     tail.prob[,k+1] <- pCopula(cbind(1 - u, 1 - u), copula = cop)
16     ## Create plot information
17     expr[k] <- as.expression(
18       substitute(group("( ", list(rho, nu), " )") ==
```

```

19                     group("(", list(RR, NN), ")"),
20                     list(RR = rho[i],
21                         NN = if(is.infinite(nu[j]))
22                             quote(infinity) else nu[j])))
23             ltys[k] <- length(rho) - i + 1
24             cols[k] <- j
25         }
26     }
27 > ## Standardize w.r.t. Gauss case
28 > tail.prob.fact <- tail.prob # for comparison to Gauss case
29 > tail.prob.fact[,2:5] <- tail.prob[,2:5] / tail.prob[,5]
30 > tail.prob.fact[,6:9] <- tail.prob[,6:9] / tail.prob[,9]
31 > ## Plot tail probabilities
32 > matplot(tail.prob[,1], tail.prob[,-1], type = "l", lwd = 2, lty = ltys,
33           col = cols, xlab = quote(P(U[1]>u, U[2]>u)^~"as a function of u"),
34           ylab = "")
35 > legend("topright", expr, bty = "n", lwd = 2, lty = ltys, col = cols)
36 > ## Plot standardized tail probabilities
37 > matplot(tail.prob.fact[,1], tail.prob.fact[,-1], log = "y", type = "l",
38           lty = ltys, col = cols, lwd = (wd <- 2*c(1,1,1,1.6,1,1,1,1)),
39           xlab = quote(P(U[1]>u, U[2]>u)^~
40                           "as a function of u standardized by Gauss case"),
41           ylab = "")
42 > legend("topleft", expr, bty = "n", lwd = wd, lty = ltys, col = cols)

```

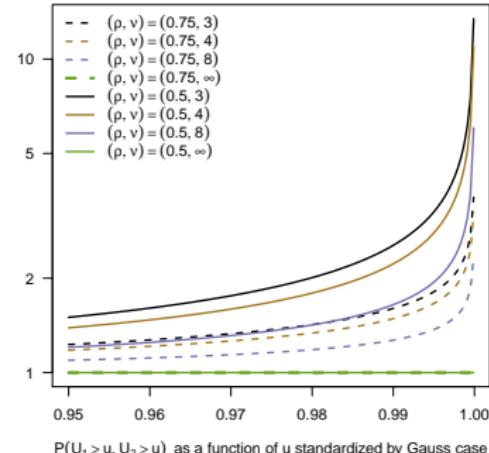
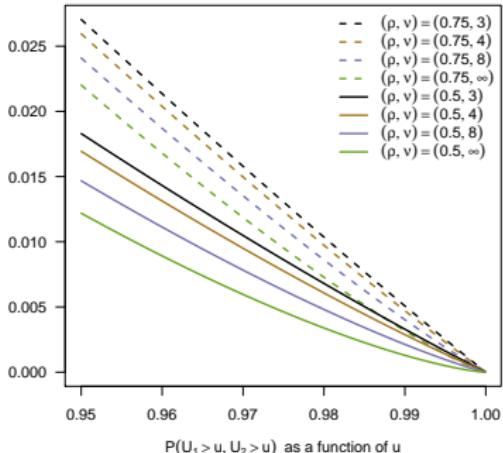


Figure 2.22 Tail probability $\mathbb{P}(U_1 > u, U_2 > u)$ as a function of u for the t copula $C_{\rho,\nu}^t$ with correlation parameter ρ and degrees of freedom ν (left) and divided by the corresponding tail probability of the normal copula (right).

- The larger ρ for fixed ν or the smaller ν for fixed ρ , the larger $u \mapsto \mathbb{P}(U_1 > u, U_2 > u)$.
- As $u \mapsto \frac{C_{\rho,\nu}^t(1-u,1-u)}{C_{\rho}^n(1-u,1-u)}$ shows, the difference between $C_{\rho,\nu}^t$ and C_{ρ}^n becomes more pronounced the smaller ρ or the smaller ν or the larger u , all other factors remaining unchanged. \square

Example 2.6.17 (Effect of ρ and ν on $d \mapsto \mathbb{P}(U_1 > 0.99, \dots, U_d > 0.99)$ for t copulas)

- This example illustrates how ρ and ν affect the (joint) tail probability $d \mapsto \mathbb{P}(U_1 > 0.99, \dots, U_d > 0.99)$ for $(U_1, \dots, U_d) \sim C_{\rho, \nu}^t$.
- By radial symmetry, $\mathbb{P}(U_1 > 0.99, \dots, U_d > 0.99) = C_{\rho, \nu}^t(0.01, \dots, 0.01)$.
- Note that, for $d \geq 3$, the t copula is evaluated by randomized quasi-Monte Carlo.

```
1 > d <- 2:20 # dimensions
2 > u <- 0.99 # level u of  $P(U_1 > u, \dots, U_d > u)$ 
3 > tail.pr.d <- matrix(d, nrow = length(d), ncol = 1+len) # tail prob; P[,1] = d
4 > set.seed(271) # set seed due to MC randomness here
5 > for(i in seq_along(rho)) { # rho
6     for(j in seq_along(nu)) { # degrees of freedom
7         k <- length(nu) * (i-1) + j
8         for(l in seq_along(d)) { # dimension
9             ## Create the copula
10            cop <- ellipCopula("t", param = rho[i], dim = d[l], df = nu[j])
11            ## Evaluate  $P(U_1 > u, \dots, U_d > u) = P(U_1 \leq 1-u, \dots, U_d \leq 1-u)$ 
12            tail.pr.d[l, k+1] <- pCopula(rep(1-u, d[l]), copula = cop)
13        }
14    }
```

```

15 }
16 > ## Standardize w.r.t. Gauss case
17 > tail.pr.d факт <- tail.pr.d # for comparison to Gauss case
18 > tail.pr.d факт[,2:5] <- tail.pr.d[,2:5] / tail.pr.d[,5]
19 > tail.pr.d факт[,6:9] <- tail.pr.d[,6:9] / tail.pr.d[,9]
20 > ## Plot tail probabilities
21 > matplot(tail.pr.d[,1], tail.pr.d[,-1], type = "l", log = "y", yaxt = "n",
22     lty = ltys, col = cols, lwd = 2, ylab = "",
23     xlab = quote(P(U[1] > 0.99, ..., U[d] > 0.99)~~
24         "as a function of d"))
25 > sfsmisc:::eaxis(2, cex.axis = 0.8)
26 > axis(1, at = 2)
27 > legend("topright", expr[1:4], bty="n", lty=ltys[1:4], col=cols[1:4], lwd=2)
28 > legend("bottomleft", expr[5:8], bty="n", lty=ltys[5:8], col=cols[5:8], lwd=2)
29 > ## Plot standardized tail probabilities
30 > matplot(tail.pr.d факт[,1], tail.pr.d факт[,-1], log = "y", type = "l",
31     las = 1, lty = ltys, col = cols,
32     lwd = (wd <- 2*c(1,1,1,1.6,1,1,1,1)), ylab = "",
33     xlab = quote(P(U[1] > 0.99,...,U[d] > 0.99)~~
34         "as a function of d standardized by Gauss case"))
35 > legend("topleft", expr, bty = "n", lty = ltys, lwd = wd, col = cols)
36 > axis(1, at = 2)

```

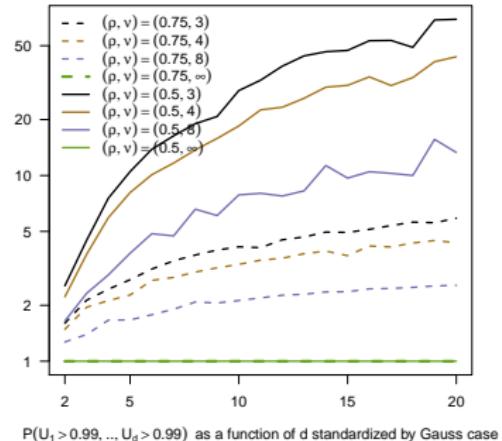
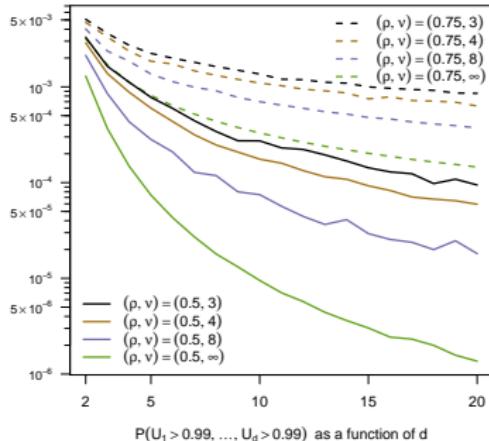


Figure 2.23 Tail probability $\mathbb{P}(U_1 > 0.99, \dots, U_d > 0.99)$ as a function of the dimension d for homogeneous t copulas with correlation parameter ρ and degrees of freedom ν (left) and divided by the corresponding tail probability for the normal copula (right).

- The larger ρ for fixed ν or the smaller ν for fixed ρ , the larger $d \mapsto \mathbb{P}(U_1 > 0.99, \dots, U_d > 0.99)$.
- As $d \mapsto \frac{C_{\rho, \nu}^t(0.01, \dots, 0.01)}{C_\rho^n(0.01, \dots, 0.01)}$ shows, the difference between C_ρ^n and $C_{\rho, \nu}^t$ becomes more pronounced the smaller ρ or the smaller ν or the larger d , all other factors remaining unchanged.

- Assume that we are given five daily negative log-returns $\mathbf{X} = (X_1, \dots, X_5)$ with fixed continuous marginal dfs and fixed common pairwise Kendall's taus equal to $1/3$. Suppose that we are unsure whether a normal or a t copula should be used as underlying dependence structure C .
- If $C = C_{0.5}^n$, the probability that, on any day, all five negative log-returns lie above their $u = 0.99$ quantiles is (up to the Monte Carlo error for $d \geq 3$) $\mathbb{P}(X_1 > F_1^\leftarrow(u), \dots, X_5 > F_5^\leftarrow(u)) = \mathbb{P}(F_1(X_1) > u, \dots, F_5(X_5) > u) = C_{0.5}^n(1 - u, \dots, 1 - u)$.
- Such an event happens about once every $1/C_{0.5}^n(1 - u, \dots, 1 - u)$ trading days on average, so once every $1/(260 C_{0.5}^n(1 - u, \dots, 1 - u))$ years.
- However, if $C = C_{0.5,3}^t$, the right-hand side of Figure 2.23 implies that such an event will happen approximately 10 times more often, that is, once every $1/(2600 C_{0.5}^n(1 - u, \dots, 1 - u))$ years (increasing in $d!$).

```
1 > ## Joint exceedance probability under the normal copula
2 > d <- 5
3 > rho <- 0.5
4 > u <- 0.99
5 > set.seed(271)
6 > ex.prob.norm <- pCopula(rep(1 - u, d), copula = normalCopula(rho, dim = d))
7 > 1 / (260 * ex.prob.norm) # ~ 51.72 years
```

[1] 51.72134

```
1 > ## Joint exceedance probability under the t copula model with 3 df
2 > ## 1) Via scaling of the probability obtained from the normal copula
3 > ## Note that the scaling factor was read off from the previous plot
4 > 1 / (2600 * ex.prob.norm) # ~ 5.17 years
```

[1] 5.172134

```
1 > ## 2) Directly using the t copula
2 > ex.prob.t3 <- pCopula(rep(1 - u, d), copula = tCopula(rho, dim = d, df = 3))
3 > 1 / (260 * ex.prob.t3) # ~ 5.91 years
```

[1] 5.911367



3 Classes and families

- 3.1 Elliptical distributions and copulas
- 3.2 Archimedean copulas
- 3.3 Extreme-value copulas
- 3.4 Some copula transformations and constructions

3.1 Elliptical distributions and copulas

- Elliptical copulas are among the most widely used copulas in practice.
- Normal (or Gauss) copulas and t copulas arise, by Sklar's Theorem, from the multivariate normal and t distribution, respectively; see Chapter 2.
- Due to their implicit construction by Sklar's Theorem, properties of elliptical copulas are typically derived from the corresponding elliptical distributions. Proofs often involve characteristic functions.
- In a nutshell, \mathbf{X} is *elliptically distributed* if $\mathbf{X} \stackrel{d}{=} \boldsymbol{\mu} + A\mathbf{Y} = \boldsymbol{\mu} + A\mathbf{R}\mathbf{S}$ with location vector $\boldsymbol{\mu} \in \mathbb{R}^d$, scale matrix $\Sigma = AA'$ with $\text{rank}(\Sigma) = k \leq d$ for a matrix $A \in \mathbb{R}^{d \times k}$ and radial part $R \geq 0$.
- Since $\Sigma = AA'$, Σ is symmetric and positive semi-definite, thus a covariance matrix ($\Sigma = \text{Cov}(A\mathbf{Z})$ for $\mathbf{Z} \sim N_k(\mathbf{0}, I_k)$). Conversely, $\Sigma = AA'$ where A is lower triangular with non-negative entries on the diagonal (*Cholesky decomposition*; A is the Cholesky factor).
- For elliptical copulas, one can assume $\boldsymbol{\mu} = \mathbf{0}$ and $\Sigma = P$ (why?)

3.2 Archimedean copulas

- An *Archimedean copula* is a copula of the form

$$C(\mathbf{u}) = \psi(\psi^{-1}(u_1) + \cdots + \psi^{-1}(u_d)), \quad \mathbf{u} \in [0, 1]^d. \quad (3.1)$$

with *generator* $\psi : [0, \infty] \rightarrow [0, 1]$, $\psi(0) = 1$, $\psi(\infty) = \lim_{t \rightarrow \infty} \psi(t) = 0$ and ψ strictly decreasing on $[0, \inf\{t : \psi(t) = 0\}]$.

- (3.1) is a *copula if and only if ψ is d -monotone* (roughly, first $d - 2$ derivatives alternate in sign). *Completely monotone ψ* (all derivatives exist and alternate in sign) are connected to dfs F on $[0, \infty)$ by *Bernstein's Theorem*: $\psi = \mathcal{LS}[F]$ or $F = \mathcal{LS}^{-1}[\psi]$.
- In terms of $V \sim F$, Archimedean copulas allow for the *stochastic representation* $\mathbf{U} = (\psi(\frac{E_1}{V}), \dots, \psi(\frac{E_d}{V})) \sim C$ where $E_1, \dots, E_d \stackrel{\text{ind.}}{\sim} \text{Exp}(1)$ are independent of V (*Marshall–Olkin algorithm*)
- Archimedean copulas are *exchangeable*, but *asymmetric extensions exist*.

3.3 Extreme-value copulas

- *Extreme-value copulas* are the copulas of multivariate extreme-value distributions. Multivariate extreme-value distributions are related to the **block maxima method**; see Gumbel (1958) or McNeil et al. (2015).
- By definition, a copula C is an **extreme-value copula** if there exists a copula C^* such that $\lim_{n \rightarrow \infty} C^*(\mathbf{u}^{1/n})^n = C(\mathbf{u})$, $\mathbf{u} \in [0, 1]^d$.
- One can show that C is an **extreme-value copula** if and only if it is **max-stable**, that is, $C(\mathbf{u}^{1/r})^r = C(\mathbf{u})$, for all $r \in \mathbb{N}$ and $\mathbf{u} \in [0, 1]^d$.
- A central result says that C is an **extreme-value copula** if and only if it can be written as

$$C(\mathbf{u}) = \exp\left(\left(\sum_{j=1}^d \log u_j\right) A\left(\frac{\log u_2}{\sum_{j=1}^d \log u_j}, \dots, \frac{\log u_d}{\sum_{j=1}^d \log u_j}\right)\right) \quad (3.2)$$

in terms of the so-called *Pickands dependence function* A .

3.4 Some copula transformations and constructions

3.4.1 Rotated copulas

- If $\mathbf{U} \sim C$, then $\mathbf{1} - \mathbf{U} \sim \bar{C}$. For $d = 2$, a sample from \bar{C} can be obtained from a sample from C by orthogonally reflecting the points at $u_1 = 1/2$ and $u_2 = 1/2$ (order arbitrary).
- This transformation can also be seen as a particular rotation.
- The definition of rotated copulas generalizes this observation. Let C be a d -dimensional copula and let $\mathbf{U} \sim C$. Given $\mathbf{r} \in \{0, 1\}^d$, we call $\text{rot}_{\mathbf{r}}(C)$ the *rotated copula* of C based on \mathbf{r} if $\mathbf{U} \sim C$ is equivalent to $((1 - r_1)U_1 + r_1(1 - U_1), \dots, (1 - r_d)U_d + r_d(1 - U_d)) \sim \text{rot}_{\mathbf{r}}(C)$. A sample from $\text{rot}_{\mathbf{r}}(C)$ can be obtained from a sample from C by “flipping the dimensions” corresponding to those components of \mathbf{r} which are equal to 1. Note that $\bar{C} = \text{rot}_{\mathbf{1}}(C)$.

- This is equivalent to adequately “rotating” the sample from C and corresponds to a sequence of orthogonal reflections with respect to the hyperplanes $\{u \in [0, 1]^d : u_j = 1/2\}$ for all j such that $r_j = 1$.
- If C admits a density c , the density $\text{rot}_r(c)$ of $\text{rot}_r(C)$ is given by

$$\text{rot}_r(c)(u) = c((1 - r_1)u_1 + r_1(1 - u_1), \dots, (1 - r_d)u_d + r_d(1 - u_d)),$$

Example 3.4.1 (A rotated bivariate Clayton copula)

Let C be C_4^C , the Clayton copula with parameter $\theta = 4$, and let $r = (1, 0)$. Rotated copulas can be created using the function `rotCopula()`. The following code produces wireframe and contour plots of $\text{rot}_{(1,0)}(C_4^C)$ and of the corresponding density $\text{rot}_{(1,0)}(c_4^C)$:

```

1 > ## The vector r is represented by a vector of logicals
2 > rc <- rotCopula(claytonCopula(4), flip = c(TRUE, FALSE))
3 > wireframe2(rc, FUN = pCopula, # wireframe plot (copula)
4   draw.4.pCoplines = FALSE)
5 > wireframe2(rc, FUN = dCopula, delta = 0.025) # wireframe plot (density)
6 > contourplot2(rc, FUN = pCopula, n.grid = 64) # contour plot (copula)
7 > contourplot2(rc, FUN = dCopula, n.grid = 64, cuts = 30,
8   pretty = FALSE, lwd = 1/2) # contour plot (density)

```

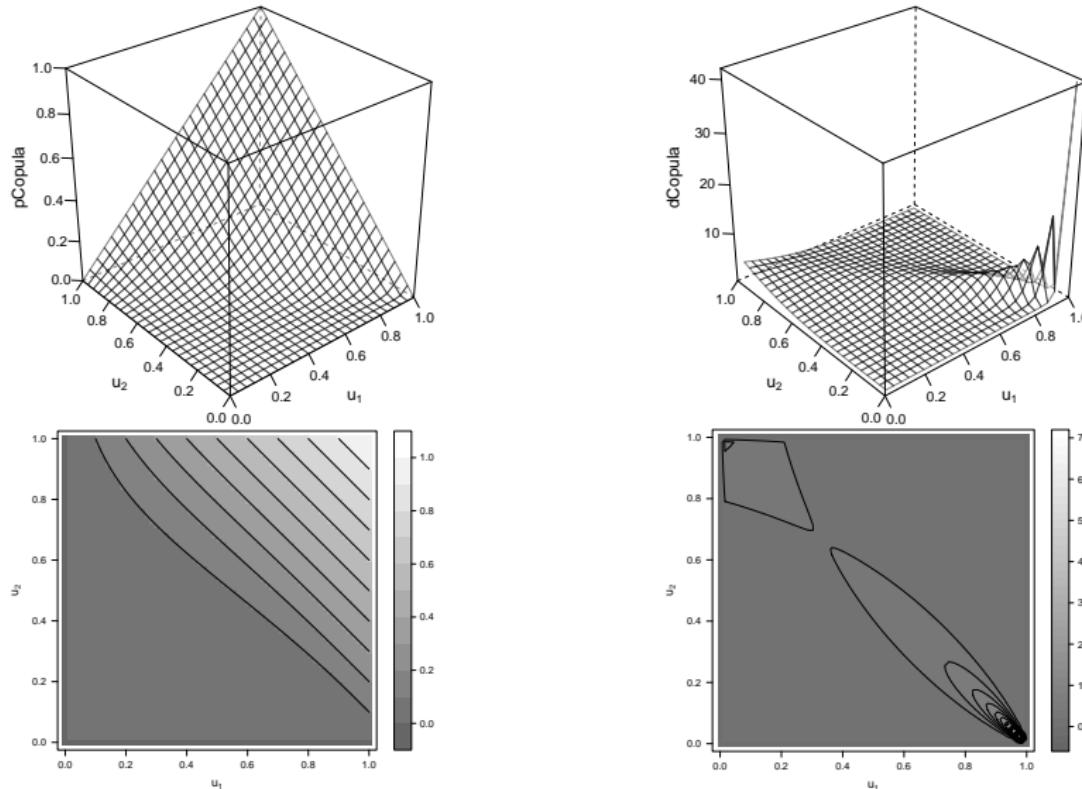


Figure 3.1 Wireframe (top) and contour plots (bottom) of $\text{rot}_{(1,0)}(C_4^C)$ (left) and of the corresponding density $\text{rot}_{(1,0)}(c_4^C)$ (right).

□

Example 3.4.2 (A rotated four-dimensional Frank copula)

Let $d = 4$, C be C_{10}^F (the Frank copula with parameter 10) and $r = (1, 0, 1, 0)$. The following code illustrates sampling from $\text{rot}_{(1,0,1,0)}(C_{10}^F)$:

```
1 > ## The logical representing vector r
2 > flip <- c(TRUE, FALSE, TRUE, FALSE)
3 > rf <- rotCopula(frankCopula(10, dim = 4), flip = flip)
4 > n <- 1000
5 > set.seed(2016)
6 > U <- rCopula(n, copula = rf)
7 > set.seed(2016)
8 > V <- rCopula(n, frankCopula(10, dim = 4))
9 > ## "Flip" the relevant components
10 > V[, flip] <- 1 - V[, flip]
11 > stopifnot(all.equal(U, V)) # check
```

The effect of r appears clearly from the scatter-plot matrix:

```
1 > splom2(U, cex = 0.3, col.mat = "black")
```

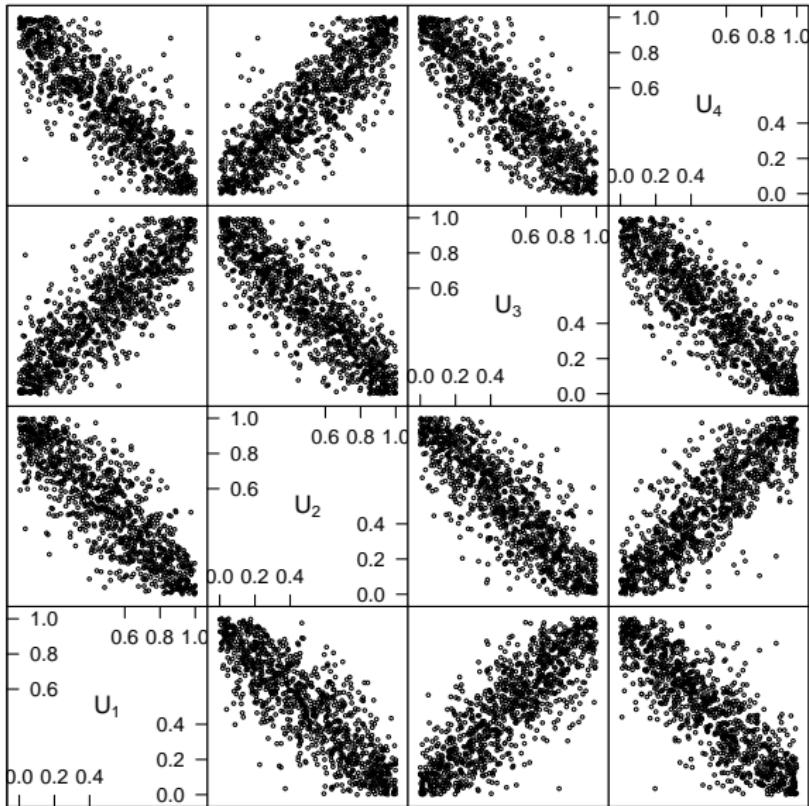


Figure 3.2 Scatter-plot matrix of $n = 1000$ independent observations from the four-dimensional rotated Frank copula $\text{rot}_{(1,0,1,0)}(C_{10}^F)$. □

- Spearman's rho and Kendall's tau of $\text{rot}_r(C)$ are $(-1)^{1(r_1 \neq r_2)} \rho_s$ and $(-1)^{1(r_1 \neq r_2)} \tau$, respectively.
- For the coefficients of tail dependence of $\text{rot}_r(C)$ for $r \neq 0$, only $r = 1$ is trivial, in which case the coefficient of upper (respectively, lower) tail dependence of $\text{rot}_1(C)$ is the coefficient of lower (respectively, upper) tail dependence of C .

3.4.2 Khoudraji's device

- Given two d -dimensional copulas C_1, C_2 and a vector $s \in [0, 1]^d$, *Khoudraji's device* $\text{kho}_s(C_1, C_2)$ with shape vector s is defined by

$$\text{kho}_s(C_1, C_2)(\mathbf{u}) = C_1(\mathbf{u}^{1-s})C_2(\mathbf{u}^s), \quad \mathbf{u} \in [0, 1]^d, \quad (3.3)$$

with the convention that $\mathbf{u}^v = (u_1^{v_1}, \dots, u_d^{v_d})$ for all $\mathbf{u}, \mathbf{v} \in [0, 1]^d$.

- This construction can lead to non-exchangeable copulas and was initially proposed by Khoudraji (1995); see Genest et al. (1998) for further details.

- It is a special case of the construction proposed by Liebscher (2008) who considers the product of copulas whose arguments are properly transformed to ensure that the product has uniform margins.
- Sampling from copulas constructed using Khoudraji's device is particularly simple. If $\mathbf{U} \sim C_1$ independently of $\mathbf{V} \sim C_2$, then

$$\left(\max\left\{ U_1^{\frac{1}{1-s_1}}, V_1^{\frac{1}{s_1}} \right\}, \dots, \max\left\{ U_d^{\frac{1}{1-s_d}}, V_d^{\frac{1}{s_d}} \right\} \right) \sim \text{kho}_s(C_1, C_2),$$

with the convention $u^{1/0} = 0$ for all $u \in [0, 1)$ and $1^{1/0} = 1$.

- A useful subset of families constructed from Khoudraji's device is obtained by taking C_1 or C_2 to be Π . For example, if $C_1 = \Pi$ and C_2 denotes a Clayton copula, one obtains a *Khoudraji–Clayton copula*.

Example 3.4.3 (Non-exchangeable Khoudraji–Clayton copulas)

Let $C_1 = \Pi$ and $C_2 = C_6^C$ (Clayton copula with parameter $\theta = 6$) and let $s = (0.2, 0.95)$. An object representing $kho_{(0.2, 0.95)}(\Pi, C_6^C)$ can be created as follows:

```
1 > (kc <- khoudrajiCopula(copula2 = claytonCopula(6), shapes = c(0.2, 0.95)))
```

```
Khoudraji copula, dim. d = 2, constructed from
Independence copula
Clayton copula
Dimension: 2
Parameters:
  c2.alpha    = 6.00
  shape1      = 0.20
  shape2      = 0.95
```

$kho_{(0.2, 0.95)}(\Pi, C_6^C)$ has three parameters: the one of $C_2 = C_6^C$ and two shape parameters.

We now empirically investigate the influence of s_1 on the asymmetry of the copula $\text{kho}_{(s_1, 0.95)}(\Pi, C_6^C)$. To this end, we generate $n = 5000$ independent observations from $\text{kho}_{(s_1, 0.95)}(\Pi, C_6^C)$ for $s_1 \in \{0.2, 0.4, 0.6, 0.8\}$.

```
1 > n <- 5000
2 > plot(rCopula(n, copula = kc),
3         cex = 0.5, xlab = quote(U[1]), ylab = quote(U[2]))
4 > plot(rCopula(n, copula = setTheta(kc, value = c(6, 0.4, 0.95))),
5         cex = 0.5, xlab = quote(U[1]), ylab = quote(U[2]))
6 > plot(rCopula(n, copula = setTheta(kc, value = c(6, 0.6, 0.95))),
7         cex = 0.5, xlab = quote(U[1]), ylab = quote(U[2]))
8 > plot(rCopula(n, copula = setTheta(kc, value = c(6, 0.8, 0.95))),
9         cex = 0.5, xlab = quote(U[1]), ylab = quote(U[2]))
```

The resulting scatter plots are displayed in Figure 3.3. As one can see, the non-exchangeability becomes clearly apparent for $s_1 \in \{0.4, 0.6, 0.8\}$. It is, however, not obvious how the three corresponding scatter plots should be ranked in terms of strength of asymmetry; see, for instance, Genest and Nešlehová (2013, Section 5.2) and the references therein for possible measures of asymmetry.

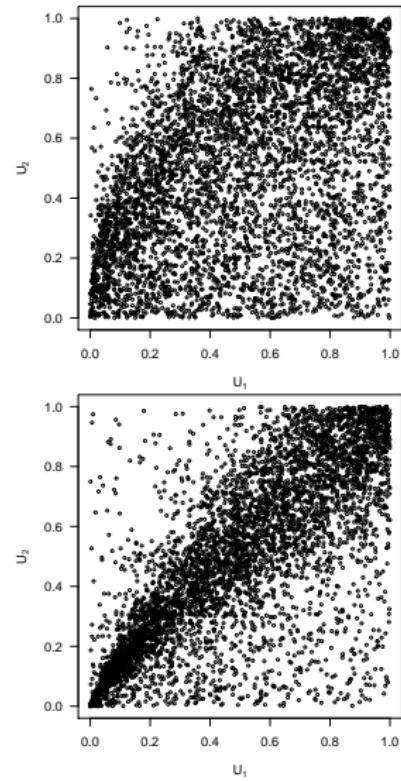
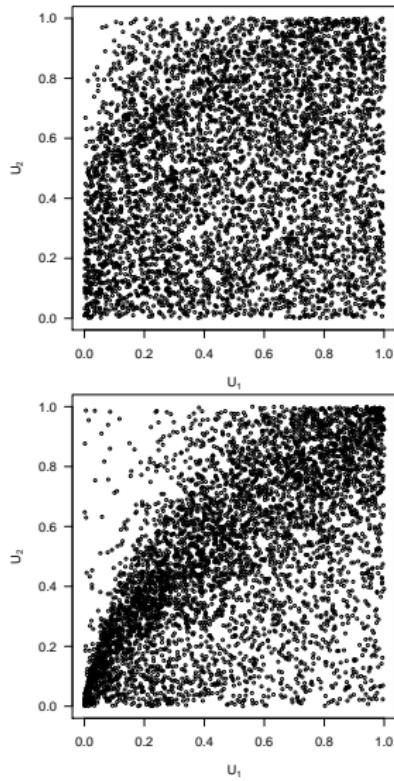


Figure 3.3 Scatter plots of $n = 5000$ independent observations from $\text{kho}_{(0.2, 0.95)}(\Pi, C_6^C)$ (top left), $\text{kho}_{(0.4, 0.95)}(\Pi, C_6^C)$ (top right), $\text{kho}_{(0.6, 0.95)}(\Pi, C_6^C)$ (bottom left) and $\text{kho}_{(0.8, 0.95)}(\Pi, C_6^C)$ (bottom right).

Exchanging the roles of s_1 and s_2 in (3.3) can be visualized as follows:

```
1 > plot(rCopula(n, copula = setTheta(kc, value = c(6, 0.95, 0.6))),  
2       cex = 0.5, xlab = quote(U[1]), ylab = quote(U[2]))
```

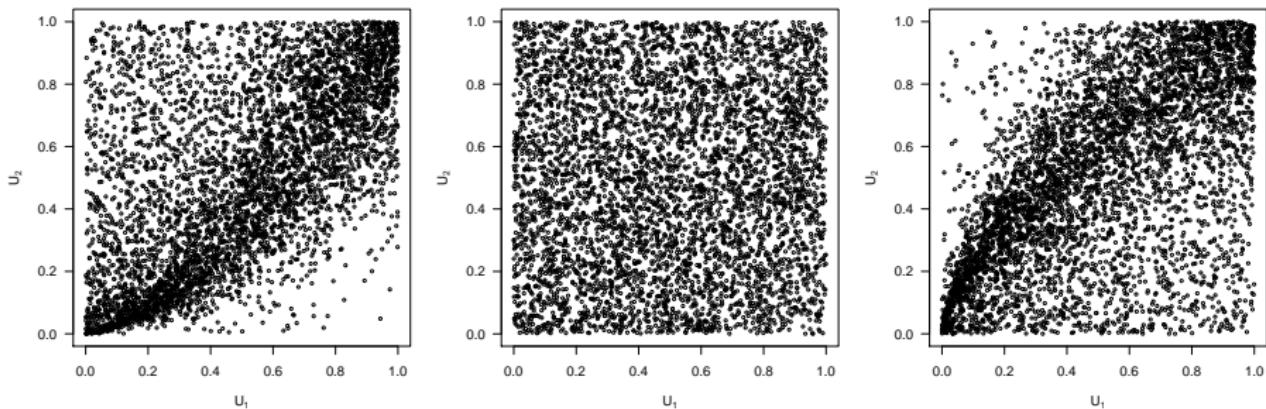


Figure 3.4 Scatter plots of $n = 5000$ independent observations from $\text{kho}_{(0.95,0.6)}(\Pi, C_6^C)$ (left), $\text{kho}_{(0.6,0.95)}(C_6^C, \Pi)$ (middle) and $\text{kho}_{(1-0.6,1-0.95)}(C_6^C, \Pi)$ (right).

Comparing the left-hand side plot with the bottom-left one of Figure 3.3 confirms that exchanging s_1 and s_2 amounts to performing the two said reflections.

The effect of exchanging the roles of Π and C_6^C (see middle plot):

```
1 > plot(rCopula(n, copula = khoudrajiCopula(copula1 = claytonCopula(6),
2                               shapes = c(0.6, 0.95))),
3       cex = 0.5, xlab = quote(U[1]), ylab = quote(U[2]))
```

$kho_{(0.6,0.95)}(C_6^C, \Pi)$ (middle plot) is close to Π . If Π is used as second copula, to obtain asymmetry, either s_1 or s_2 need to be close to 0. As the right-hand side plot confirms, $kho_{(1-0.6,1-0.95)}(C_6^C, \Pi) = kho_{(0.6,0.95)}(\Pi, C_6^C)$:

```
1 > plot(rCopula(n, copula = khoudrajiCopula(copula1 = claytonCopula(6),
2                               shapes = c(1 - 0.6, 1 - 0.95))),
3       cex = 0.5, xlab = quote(U[1]), ylab = quote(U[2]))
```

□

Using Π for C_1 or C_2 in (3.3) can lead to a limited range of attainable positive associations. For $d = 2$, Genest et al. (2011a) showed that Kendall's tau τ of $kho_s(\Pi, C)$ with $s_1, s_2 \in (0, 1)$ satisfies

$$\tau \leq \frac{s_1 s_2}{s_1 + s_2 - s_1 s_2}. \quad (3.4)$$

For instance, if $s_1 = 0.6$ and $s_2 = 0.95$, no matter how C is chosen, τ will not be able to exceed, approximately, 0.5816.

Example 3.4.4 (Non-exchangeable Khoudraji–Gumbel–Hougaard—Clayton copulas)

Let $d = 2$, C_1 be the Gumbel–Hougaard copula $C_{\theta_1}^{\text{GH}}$ and C_2 be the Clayton copula $C_{\theta_2}^C$. Choose θ_1, θ_2 such that Kendall's tau of both C_1 and C_2 is equal to $\tau \in \{0.65, 0.7, 0.95\}$. For each value of τ , generate $n = 5000$ independent observations from $\text{kho}_{(0.6, 0.95)}(C_{\theta_1}^{\text{GH}}, C_{\theta_2}^C)$:

```
1 > ## The setup
2 > s <- c(0.6, 0.95)
3 > copula1 <- gumbelCopula
4 > copula2 <- claytonCopula
5 > ## A utility function to obtain the parameter values of C_1 and C_2
6 > param <- function(tau) c(iTau(copula1(), tau), iTau(copula2(), tau))
7 > ## The corresponding Khoudraji–Gumbel–Hougaard–Clayton copula
8 > (kho <- khoudrajiCopula(copula1 = copula1(param(0.65)[1]),
9 >                           copula2 = copula2(param(0.65)[2]),
10 >                          shapes = s))
```

```
Khoudraji copula, dim. d = 2, constructed from
```

```
Gumbel copula
```

```
Clayton copula
```

```
Dimension: 2
```

```
Parameters:
```

```
c1.alpha = 2.857143
```

```
c2.alpha = 3.714286
```

```
shape1 = 0.600000
```

```
shape2 = 0.950000
```

```
1 > n <- 5000
2 > U <- rCopula(n, copula = kho)
3 > plot(U, cex = 0.5, xlab = quote(U[1]), ylab = quote(U[2]))
4 > V <- rCopula(n, copula = setTheta(kho, value = c(param(0.8), s)))
5 > plot(V, cex = 0.5, xlab = quote(V[1]), ylab = quote(V[2]))
6 > W <- rCopula(n, copula = setTheta(kho, value = c(param(0.95), s)))
7 > plot(W, cex = 0.5, xlab = quote(W[1]), ylab = quote(W[2]))
```

The produced scatter plots are displayed in Figure 3.5.

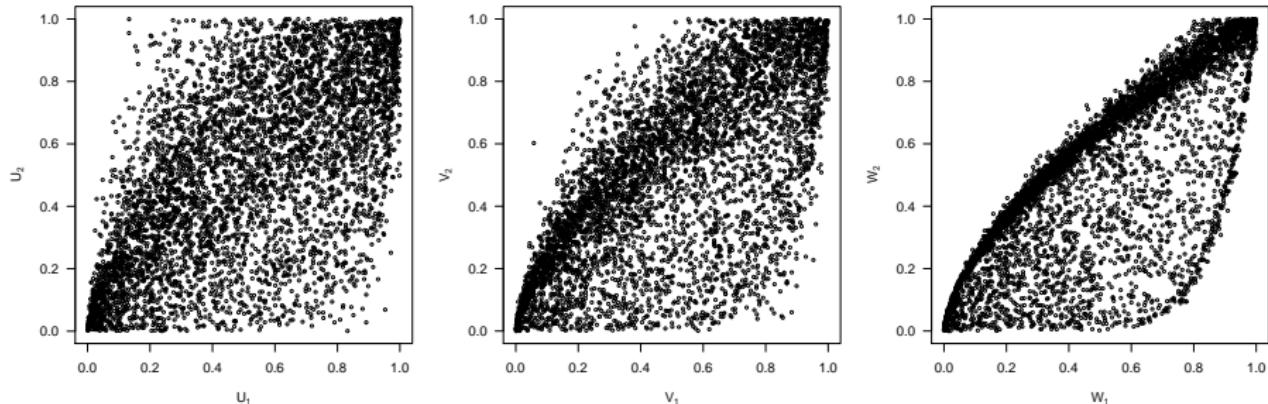


Figure 3.5 Scatter plots of $n = 5000$ independent observations from $\text{kho}_{(0.6, 0.95)}(C_{\theta_1}^{\text{GH}}, C_{\theta_2}^{\text{C}})$ with θ_1 and θ_2 chosen so that $C_{\theta_1}^{\text{GH}}$ and $C_{\theta_2}^{\text{C}}$ both have a Kendall's tau of 0.65 (left), 0.8 (middle) and 0.95 (right).

Estimate Kendall's tau from all three data sets from $\text{kho}_{(0.6, 0.95)}(C_{\theta_1}^{\text{GH}}, C_{\theta_2}^{\text{C}})$:

```
1 > c(cor(U, method = "kendall")[1,2], cor(V, method = "kendall")[1,2],
2   cor(W, method = "kendall")[1,2])
```

```
[1] 0.4382486 0.5049387 0.6342441
```

It is thus possible to choose C_1 to exceed the bound 0.5816 for Kendall's tau related to (3.4). □

- Khoudraji's device can also be used to construct non-exchangeable extreme-value copula families.
- If C_1 and C_2 in (3.3) are extreme-value copulas, so is $\text{kho}_s(C_1, C_2)$ for all $s \in [0, 1]^d$.
- $C_1 = \Pi$ leads to relatively simple non-exchangeable extreme-value copulas.

Example 3.4.5 (A non-exchangeable extreme-value family)

Let $d = 2$, $C_2 = C_4^{\text{GH}}$ and $s_2 = 0.95$. The following code illustrates the influence of s_1 on the Pickands dependence function of $\text{kho}_{(s_1, 0.95)}(\Pi, C_4^{\text{GH}})$:

```

1 > kg <- khoudrajiCopula(copula2 = gumbelCopula(4), shapes = c(0.2, 0.95))
2 > curve(A(kg, w = x), from = 0, to = 1, ylim = c(0.5, 1),
3       xlab = "t", ylab = expression({A[theta]^{\{KGH\}}}(t)), col = 1, lwd = 2)
4 > curve(A(setTheta(kg, value = c(4, 0.4, 0.95)), w = x), add = TRUE, lwd = 2,
5       col = 2)
6 > curve(A(setTheta(kg, value = c(4, 0.6, 0.95)), w = x), add = TRUE, lwd = 2,
7       col = 3)
8 > curve(A(setTheta(kg, value = c(4, 0.8, 0.95)), w = x), add = TRUE, lwd = 2,
9       col = 4)
10 > lines(c(0, 1),      c(1, 1),      lty = 2)

```

```

11 > lines(c(0, 0.5, 1), c(1, 0.5, 1), lty = 2)
12 > legend("bottomright", bty = "n", lwd = 2,, col = 1:4,
13     legend = expression(s[1] == 0.2, s[1] == 0.4,
14     s[1] == 0.6, s[1] == 0.8))

```

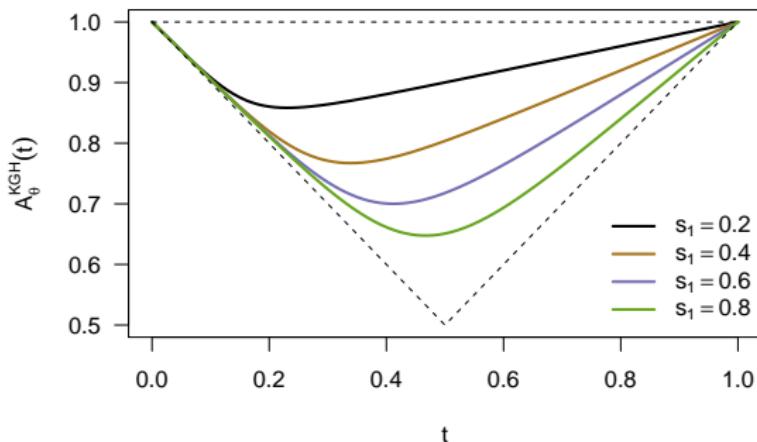


Figure 3.6 Pickands dependence function of the bivariate Khoudraji–Gumbel–Hougaard copula $\text{kho}_{(s_1, 0.95)}(\Pi, C_4^{\text{GH}})$ for $s_1 \in \{0.2, 0.4, 0.6, 0.8\}$.

The larger s_1 , the closer the Pickands dependence function is to $t \mapsto \max\{1-t, t\}$ (the one of M). It appears that, the stronger the dependence, the less room for non-exchangeability. □

- Density evaluation for copulas constructed using Khoudraji's device becomes challenging as d increases.
- For implementation reasons, when $d > 2$, density evaluation in copula is possible when C_1 and C_2 have explicit density expressions.

Example 3.4.6 (Higher-dimensional Khoudraji copulas)

Consider a Khoudraji–Gumbel–Hougaard–survival Clayton copula. The density expressions of the Gumbel–Hougaard and the survival Clayton copula being explicit, the density of the corresponding Khoudraji copula can be evaluated:

```
1 > kgsc <- khoudrajiCopula(copula1 = gumbelCopula(2, dim=3),
2                               copula2 = rotCopula(claytonCopula(6, dim=3)),
3                               shapes = c(.6, 0.7, 0.95))
4 > ## Random points in the unit hypercube where to evaluate the density
5 > set.seed(42)
6 > v <- matrix(runif(15), 5, 3)
7 > dCopula(v, copula = kgsc)
```

```
[1] 0.2852876148 0.7332486257 0.0007894093 0.6996430796 0.8508682765
```

However, if, for instance, a **normal copula** is used as second copula, density evaluation of the resulting Khoudraji copula becomes **unavailable**:

```
1 > kgn <- khoudrajiCopula(copula1 = gumbelCopula(2, dim=3),  
2                               copula2 = normalCopula(0.9, dim=3),  
3                               shapes = c(.6, 0.7, 0.95))  
4 > try( dCopula(v, copula = kgn) ) # not implemented
```

```
Error in (function (classes, fdef, mtable) :  
  unable to find an inherited method for function 'dCopula' for signature '"matrix", "  
  khoudrajiCopula"'
```



Depending on the implementation of the density expressions, the **dimension for supported Archimedean copulas is limited**. Also, **density evaluation** (and thus fitting) **may be affected by numerical issues**, for example, near the boundary of the unit hypercube; see Yan (2007, Section 5).

4 Estimation

- 4.1 Estimation under a parametric assumption on the copula
- 4.2 Non-parametric estimation of the copula

- Assume that one has a *random sample* $\mathbf{X}_1, \dots, \mathbf{X}_n$ (that is, independent copies) of $\mathbf{X} \sim H$ with continuous margins F_1, \dots, F_d . By Sklar's Theorem there exists a *unique copula* C on $[0, 1]^d$, such that

$$H(\mathbf{x}) = C(F_1(x_1), \dots, F_d(x_d)), \quad \mathbf{x} \in \mathbb{R}^d. \quad (4.1)$$

- Consider the situation in which F_1, \dots, F_d are all *unknown*.
- We now address the *estimation* of C from $\mathbf{X}_1, \dots, \mathbf{X}_n$ in this setting.

4.1 Estimation under a parametric assumption on the copula

- Assume that C in (4.1) belongs to an *absolutely continuous parametric family* of copulas

$$\mathcal{C} = \{C_\theta : \theta \in \Theta\},$$

where $\Theta \subseteq \mathbb{R}^p$, $p \geq 1$, is the parameter space.

- Assuming that $C \in \mathcal{C}$ is equivalent to assuming that there exists $\theta_0 \in \Theta$ such that $C = C_{\theta_0}$. Our goal is to estimate the true, but unknown θ_0 .
- If the (unknown) margins F_1, \dots, F_d were known, the sample

$$\mathbf{U}_i = (F_1(X_{i1}), \dots, F_d(X_{id})), \quad i \in \{1, \dots, n\}, \quad (4.2)$$

would be observable, iid, and, from Lemma 2.4.1, a sample from C .

- In that case, θ_0 could be estimated using classical techniques for iid data such as maximum likelihood estimation.
- However, since the margins of H are unknown, the random sample (4.2) is not observable, and the margins are nuisance parameters that have to be estimated so that θ_0 can be estimated.
- One can distinguish two approaches for estimating F_1, \dots, F_d : parametric and non-parametric. In the former case, the resulting estimate of C_{θ_0} will be fully parametric, while in the latter case, it will be a semi-parametric estimate.

4.1.1 Parametrically estimated margins

- Assume that F_1, \dots, F_d belong to **absolutely continuous parametric families** of univariate dfs $\mathcal{F}_1, \dots, \mathcal{F}_d$, respectively, where, for any $j \in \{1, \dots, d\}$,

$$\mathcal{F}_j = \{F_{j,\gamma_j} : \gamma_j \in \Gamma_j\},$$

and $\Gamma_j \subseteq \mathbb{R}^{p_j}$ for some integer $p_j \geq 1$. In other words, assume that, for any $j \in \{1, \dots, d\}$, there exists $\gamma_{0,j} \in \Gamma_j$ such that $F_j = F_{j,\gamma_{0,j}}$.

- Our goal is to **estimate C_{θ_0}** from $\mathbf{X}_1, \dots, \mathbf{X}_n$ under the assumption that the unknown df H belongs to the parametric family of dfs $\mathcal{H} = \{C(F_1(\cdot), \dots, F_d(\cdot)) : C \in \mathcal{C} \text{ and } F_j \in \mathcal{F}_j \text{ for all } j \in \{1, \dots, d\}\}$.

4.1.1.1 Maximum likelihood estimator

- A straightforward approach is to **estimate $(\gamma_{0,1}, \dots, \gamma_{0,d}, \theta_0)$** and obtain an estimate of θ_0 as a by-product.

- By absolute continuity of \mathcal{C} and $\mathcal{F}_1, \dots, \mathcal{F}_d$, \mathcal{H} is absolutely continuous. Differentiating Sklar's representation (4.1) of $H \in \mathcal{H}$ implies that the corresponding density h is given by

$$h(\mathbf{x}) = c_{\boldsymbol{\theta}}(F_{1,\gamma_1}(x_1), \dots, F_{d,\gamma_d}(x_d)) \prod_{j=1}^d f_{j,\gamma_j}(x_j),$$

where $c_{\boldsymbol{\theta}}$ is the density of $C_{\boldsymbol{\theta}}$ and f_{j,γ_j} is the density of F_{j,γ_j} , $j \in \{1, \dots, d\}$; see also 4) in Remark 2.3.2.

- The parameter vector $(\gamma_{0,1}, \dots, \gamma_{0,d}, \boldsymbol{\theta}_0)$ could then be estimated by the *maximum likelihood estimator (MLE)*, that is, the maximizer $(\gamma_{n,1}, \dots, \gamma_{n,d}, \boldsymbol{\theta}_n)$ of the **log-likelihood** function

$$\begin{aligned} \ell_n(\gamma_1, \dots, \gamma_d, \boldsymbol{\theta}) &= \sum_{i=1}^n \log c_{\boldsymbol{\theta}}(F_{1,\gamma_1}(X_{i1}), \dots, F_{d,\gamma_d}(X_{id})) \\ &\quad + \sum_{i=1}^n \sum_{j=1}^d \log f_{j,\gamma_j}(X_{ij}). \end{aligned} \tag{4.3}$$

- The MLE is theoretically well-grounded, see, for instance, Lehmann and Casella (1998), and provides not only the estimate C_{θ_n} of C but also the estimate $F_{j,\gamma_{n,j}}$ of F_{j,γ_j} , $j \in \{1, \dots, d\}$, and thus the estimate $C_{\theta_n}(F_{1,\gamma_{n,1}}(\cdot), \dots, F_{d,\gamma_{n,d}}(\cdot))$ of H .

Example 4.1.1 (Estimation of copula parameters via the MLE)

- Assume \mathcal{C} is the bivariate Clayton family, \mathcal{F}_1 is the family of univariate normal dfs $N(\mu, \sigma^2)$ and \mathcal{F}_2 is the family of exponential dfs $Exp(\lambda)$.
- The following code illustrates the estimation of the Clayton parameter θ_0 from a sample of $n = 1000$ independent observations from such a distribution with marginal parameters $\gamma_{0,1} = (\mu_0, \sigma_0) = (0, 1)$ and $\gamma_{0,2} = \lambda_0 = 1$, and copula parameter $\theta_0 = 3$.

```

1 > ## The "unknown" copula (a 2-dim. Clayton copula with parameter 3)
2 > cc <- claytonCopula(3)
3 > ## The "unknown" distribution (N(0,1), Exp(1) margins)
4 > mcc <- mvdc(cc, margins = c("norm", "exp"),
5   paramMargins = list(list(mean = 0, sd = 1),
6     list(rate = 1)))
7 > ## Generate the "observed" sample

```

```
8 > set.seed(712)
9 > n <- 1000
10 > X <- rMvdc(n, mvdc = mcc)
11 > ## The function fitMvdc() estimates all the parameters of the mvdc object
12 > ## mcc (whose parameter values are not used). Starting values need to be
13 > ## provided.
14 > start <- c(mu0 = mean(X[,1]), sig0 = sd(X[,1]), lam0 = 1 / mean(X[,2]),
15 > th0 = 2)
16 > (mle <- fitMvdc(X, mvdc = mcc, start = start))
```

```
Call: fitMvdc(data = X, mvdc = mcc, start = start)
Maximum Likelihood estimation based on 1000 2-dimensional observations.
Copula: claytonCopula
Margin 1 :
  m1.mean    m1.sd
-0.02231  0.99149
Margin 2 :
  m2.rate
    1.014
Copula:
  alpha
  3.117
The maximized loglikelihood is -1760
Optimization converged
```

Standard errors for the marginal and copula parameter estimates:

```
1 > summary(mle)
```

```
Call: fitMvdc(data = X, mvdc = mcc, start = start)
Maximum Likelihood estimation based on 1000 2-dimensional observations.
Copula: claytonCopula
Margin 1 :
      Estimate Std. Error
m1.mean -0.02231    0.026
m1.sd     0.99149    0.013
Margin 2 :
      Estimate Std. Error
m2.rate   1.014     0.031
Clayton copula, dim. d = 2
      Estimate Std. Error
alpha     3.117     0.125
The maximized loglikelihood is -1760
Optimization converged
Number of loglikelihood evaluations:
function gradient
        49          10
```



- Maximum likelihood estimation suffers from two drawbacks:
 - 1) The parametric assumptions made on F_1, \dots, F_d (if the margins are partially misspecified, that is, if, for some $j \in \{1, \dots, d\}$, $F_j \notin \mathcal{F}_j$, the estimation of θ_0 will be biased).
 - 2) The log-likelihood function given in (4.3) has to be maximized over a potentially high-dimensional parameter space (for instance, if $d = 10$, \mathcal{C} is the normal copula family and $\mathcal{F}_1 = \dots = \mathcal{F}_{10}$ is the family of univariate gamma dfs, the dimension of the parameter space of \mathcal{H} is $\binom{10}{2} + 10 \cdot 2 = 65$).

4.1.1.2 Inference functions for margins estimator

- The computational burden potentially associated with maximum likelihood estimation can be decreased by employing a two-stage estimator known as the *inference functions for margins estimator (IFME)*.

- First estimate the marginal parameter vectors $\gamma_{0,1}, \dots, \gamma_{0,d}$ by

$$\gamma_{n,j} = \operatorname{argsup}_{\gamma_j \in \Gamma_j} \sum_{i=1}^n \log f_{j,\gamma_j}(X_{ij}), \quad j \in \{1, \dots, d\}. \quad (4.4)$$

- Second, estimate the copula parameter vector θ_0 by

$$\theta_n = \operatorname{argsup}_{\theta \in \Theta} \sum_{i=1}^n \log c_\theta(F_{1,\gamma_{n,1}}(X_{i1}), \dots, F_{d,\gamma_{n,d}}(X_{id})).$$

- The computational advantage comes at the price of an efficiency loss relative to the maximum likelihood estimator; see Joe (2005) for the theoretical properties of the IFME.
- One can view the IFME as first forming the sample

$$U_{i,\gamma_n} = (F_{1,\gamma_{n,1}}(X_{i1}), \dots, F_{d,\gamma_{n,d}}(X_{id})), \quad i \in \{1, \dots, n\}, \quad (4.5)$$

where $\gamma_n = (\gamma_{n,1}, \dots, \gamma_{n,d})$, and then estimating θ_0 by maximizing a log-likelihood-like function, that is,

$$\theta_n = \operatorname{argsup}_{\theta \in \Theta} \sum_{i=1}^n \log c_\theta(U_{i,\gamma_n}). \quad (4.6)$$

- If the margins are correctly specified, U_{i,γ_n} , $i \in \{1, \dots, n\}$, can be viewed as a consistently estimated version of the unobservable iid \mathbf{U}_i 's.
- One thus refers to $U_{1,\gamma_n}, \dots, U_{n,\gamma_n}$ as *(parametric) pseudo-observations*; as it is not a true sample from C , θ_n in (4.6) is not a proper MLE.

Example 4.1.2 (Estimation of copula parameters via the IFME)

```

1 > ## Parametric pseudo-observations obtained from X by marginal MLE
2 > U <- cbind(pnorm(X[,1], mean = mean(X[,1])),
3                 sd = sqrt((n - 1) / n) * sd(X[,1])),
4                 pexp(X[,2], rate = 1 / mean(X[,2])))
5 > ifme <- fitCopula(claytonCopula(), data = U, method = "ml")
6 > summary(ifme)

```

```

Call: fitCopula(copula, data = data, method = "ml")
Fit based on "maximum likelihood" and 1000 2-dimensional observations.
Clayton copula, dim. d = 2
    Estimate Std. Error
alpha     3.062     0.117
The maximized loglikelihood is 637.7
Optimization converged
Number of loglikelihood evaluations:
function gradient
      5      5

```

- The standard error(s) will be consistently estimated if the data passed as argument are a true sample from the unknown copula.
- For parametric pseudo-observations and in practice, this is not the case.
- Since the estimation errors for the marginal parameters are not taken into account, the standard error computed will underestimate on average the true unknown standard error (and should not be used in inferences).
- The optimization method for computing (4.6) is controlled by optim.method of fitCopula(). It is specified (by default) with optimMeth().
- In the previous code, optim.method was automatically set to:

```
1 > optimMeth(claytonCopula(), method = "ml", dim = 2)
[1] "L-BFGS-B"
```

- The IFM method has the same drawback as MLE: If the margins are partially misspecified, the estimation of θ_0 will be biased.



4.1.2 Non-parametrically estimated margins

- To avoid misspecification of the margins one could estimate F_1, \dots, F_d non-parametrically. A typical approach is to estimate F_j , $j \in \{1, \dots, d\}$, by

$$F_{n,j}(x) = \frac{1}{n+1} \sum_{i=1}^n \mathbf{1}(X_{ij} \leq x), \quad x \in \mathbb{R}. \quad (4.7)$$

- The estimated margins are then typically used to form the sample

$$\mathbf{U}_{i,n} = (F_{n,1}(X_{i1}), \dots, F_{n,d}(X_{id})), \quad i \in \{1, \dots, n\}, \quad (4.8)$$

from which θ_0 will be estimated.

- This sample can be regarded as a consistently estimated version of the unobservable iid $\mathbf{U}_1, \dots, \mathbf{U}_n$ and is therefore referred to as a sample of *pseudo-observations* from C .

- Notice that not only $\mathbf{U}_{1,n}, \dots, \mathbf{U}_{n,n}$ are not true observations from C but $\mathbf{U}_{1,n}, \dots, \mathbf{U}_{n,n}$ are not independent (because $F_{n,j}$ depends on the j th component sample X_{1j}, \dots, X_{nj} , $j \in \{1, \dots, d\}$).
- The reason for the scaling factor $n+1$ is to ensure that the sample (4.8) lies in $(0, 1)^d$; this is important when carrying out maximum pseudo-likelihood estimation, since several copula families have non-finite density on the edges of $[0, 1]^d$ ending at $(1, \dots, 1)$.
- For any $j \in \{1, \dots, d\}$, let R_{ij} be the rank of X_{ij} among X_{1j}, \dots, X_{nj} . It is easy to verify that $F_{n,j}(X_{ij}) = R_{ij}/(n+1)$, so

$$\mathbf{U}_{i,n} = \frac{1}{n+1}(\mathbf{R}_{i1}, \dots, \mathbf{R}_{id}), \quad i \in \{1, \dots, n\}, \quad (4.9)$$

(and $U_{ij,n} = R_{ij}/(n+1)$, $i \in \{1, \dots, n\}$, $j \in \{1, \dots, d\}$).

Example 4.1.3 (Pseudo-observations of daily log-returns)

Consider daily log-return data for the period 1996–2000 for Intel, Microsoft and General Electric; see McNeil et al. (2015, Chapter 7). The small number of ties (incompatible with continuous margins) and the weak serial dependence (stylized fact of financial time series) is ignored here:

```
1 > data(rdj) # 'head(rdj)' for looking at the first six observations  
2 > spлом2(rdj[,2:4], cex = 0.4, col.mat = adjustcolor("black", 0.5))
```

The pseudo-observations can be computed via `pobs()`:

```
1 > U <- pobs(rdj[,2:4])  
2 > spлом2(U, cex = 0.4, col.mat = "black")
```



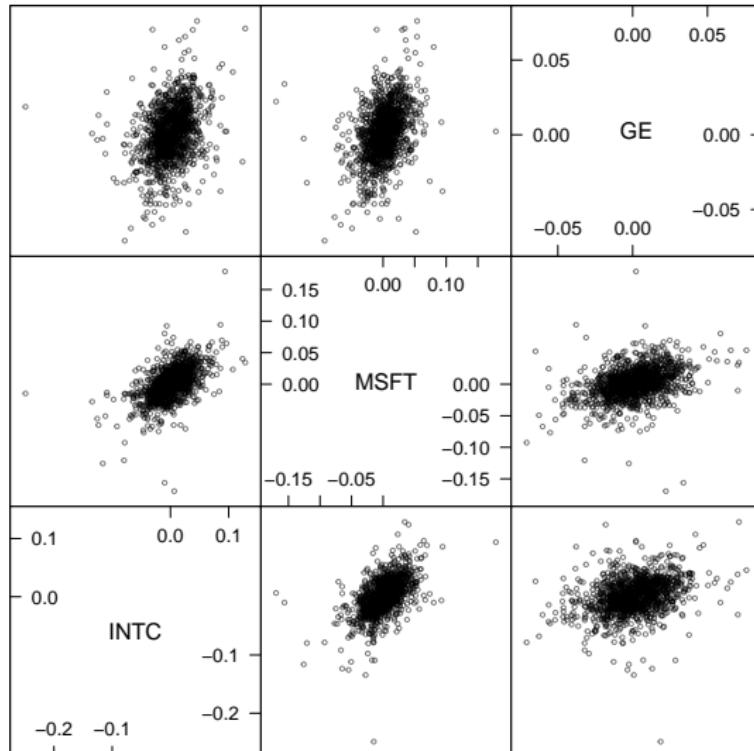


Figure 4.1 Scatter-plot matrix of the rdj data set.

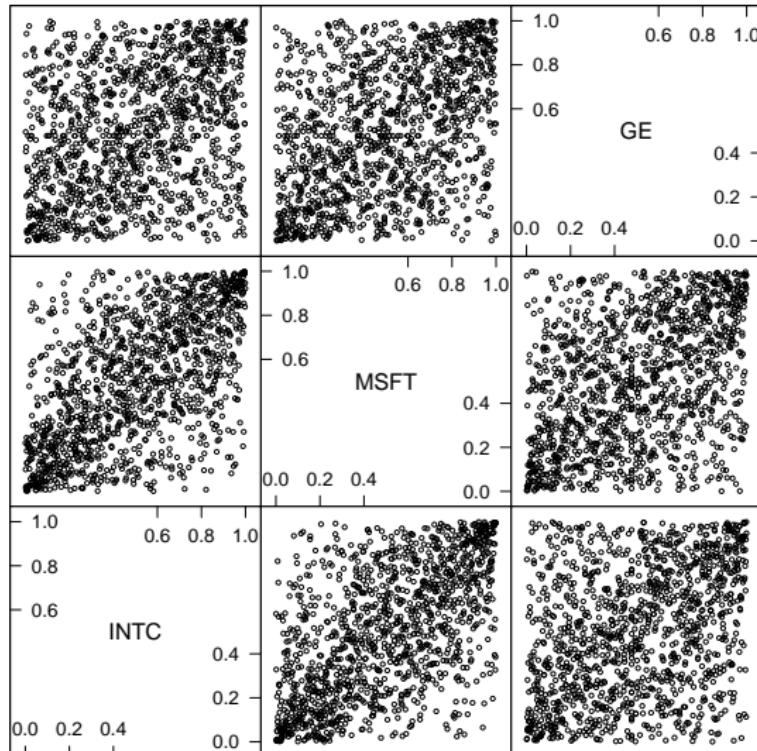


Figure 4.2 Pseudo-observations corresponding to Figure 4.1.

Two popular estimation methods based on non-parametrically estimated margins are:

- method-of-moments approaches (Kendall's tau or Spearman's rho);
- maximum pseudo-likelihood estimation.

Because they are solely based on the pseudo-observations, they are margin-free (invariant under strictly increasing transformations of the margins).

4.1.2.1 Method-of-moments estimators

- Method-of-moments estimators in the copula context are direct transpositions of well-known method-of-moments estimators. Moments of random variables are replaced by moments of the copula such as Kendall's tau or Spearman's rho.
- Although it is possible to use method-of-moments estimators when $d \geq 3$ or $p \geq 2$, we first consider the case of a bivariate sample and a one-parameter copula family $\mathcal{C} = \{C_\theta : \theta \in \Theta\}$.

- Given \mathcal{C} , let

$$g_\tau(\theta) = \tau(C_\theta) \quad \text{and} \quad g_{\rho_s}(\theta) = \rho_s(C_\theta), \quad \theta \in \Theta \subseteq \mathbb{R},$$

where $\rho_s(C_\theta)$ and $\tau(C_\theta)$ are defined in (2.18) and (2.19).

- Method-of-moments estimators based on Kendall's tau (respectively, Spearman's rho) can be used for the family \mathcal{C} if the function g_τ (respectively, g_{ρ_s}) is one-to-one:

$$\theta_n = g_\tau^{-1}(\tau_n) \quad (\text{respectively, } \theta_n = g_{\rho_s}^{-1}(\rho_{s,n}));$$

τ_n (respectively, $\rho_{s,n}$) is the sample version of Kendall's tau (respectively, Spearman's rho) computed from the pseudo-observations.

- Standard errors can be computed as explained in Genest and Favre (2007, p. 353) or Kojadinovic and Yan (2010a, Section 2.1).
- When $d > 2$ and the family \mathcal{C} is exchangeable and one-parametric one can apply g_τ^{-1} (respectively, $g_{\rho_s}^{-1}$) to the average of the sample Kendall's taus (respectively, Spearman's rhos) of the $\binom{d}{2}$ different bivariate margins.

Example 4.1.4 (Estimation of copula parameters via the method of moments based on Kendall's tau)

- Consider \mathcal{C} to be the Gumbel-Hougaard copula family with $g_\tau(\theta) = 1 - 1/\theta$ and $\theta_n = g_\tau^{-1}(\tau_n) = 1/(1 - \tau_n)$.
 - The following code illustrates the estimation of θ_0 for a Gumbel-Hougaard copula with parameter $\theta_0 = 3$ and $N(0, 1)$ margins:

[1] 2.976439

```
1 > stopifnot(all.equal(itau, 1 / (1 - tau.n))) # the same
2 > ## The same but with a standard error
3 > summary(fitCopula(gumbelCopula(), data = pobs(X), method = "itau"))
```

```
Call: fitCopula(copula, data = data, method = "itau")
Fit based on "inversion of Kendall's tau" and 1000 2-dimensional observations.
Gumbel copula, dim. d = 2
    Estimate Std. Error
alpha     2.976     0.114
```

- The choice of the margins is irrelevant since Kendall's tau (being rank-based) is invariant under strictly increasing marginal transformations.
- Note that the standard error returned by `summary(fitCopula(, method = "itau"))` is computed assuming that the input data are pseudo-observations; see Kojadinovic and Yan (2010a, Section 2.1).
- Consequently, the argument `data` of `fitCopula()` needs to be pseudo-observations for the standard error to be consistently estimated.



Example 4.1.5 (Estimation of copula parameters via the method of moments based on Spearman's rho)

- Consider \mathcal{C} to be the normal copula family with $g_{\rho_s}(\theta) = 6/\pi \arcsin(\theta/2)$ and $\theta_n = g_{\rho_s}^{-1}(\rho_{s,n}) = 2 \sin(\pi \rho_{s,n}/6)$.
- The following code illustrates the estimation of θ_0 for any meta-normal distribution with copula parameter $\theta_0 = 0.5$ and continuous margins:

```
1 > ## The "unknown" copula (a 2-dim. normal copula with parameter 0.5)
2 > nc <- normalCopula(0.5)
3 > ## Generate the "observed" sample
4 > set.seed(314)
5 > X <- rCopula(1000, nc)
6 > ## The sample estimate of Spearman's rho
7 > rho.n <- cor(X[,1], X[,2], method = "spearman")
8 > ## The corresponding copula parameter estimate
9 > (irho <- iRho(nc, rho = rho.n))
```

```
[1] 0.5498842
```

```
1 > stopifnot(all.equal(irho, 2 * sin(pi * rho.n / 6))) # the same
2 > ## The same but with a standard error
3 > summary(fitCopula(normalCopula(), data = pobs(X), method = "irho"))
```

```
Call: fitCopula(copula, data = data, method = "irho")
Fit based on "inversion of Spearman's rho" and 1000 2-dimensional observations.
Normal copula, dim. d = 2
    Estimate Std. Error
rho.1    0.5499     0.024
```



Example 4.1.6 (Application of the estimation via the method of moments based on Kendall's tau)

- Consider the [danube](#) data set of [lcopula](#).
- It consists of bivariate pseudo-observations of preprocessed [monthly average flow rates for two stations](#) (one situated at Scharding (Austria) on the Inn river and one at Nagymaros (Hungary) [on the Danube](#)).

```
1 > data(danube, package = "lcopula") # already pseudo-observations
2 > U <- as.matrix(danube)
3 > plot(U, xlab = "Donau", ylab = "Inn")
```

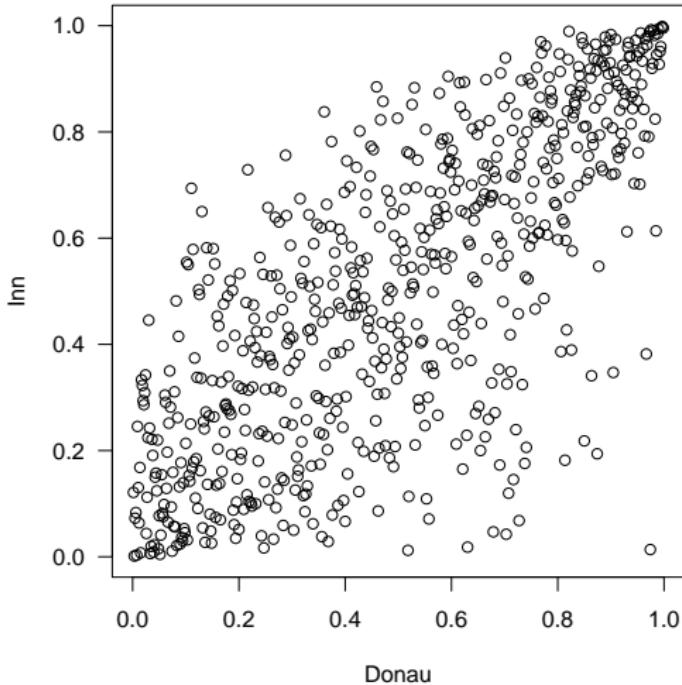


Figure 4.3 Scatter plot of the pseudo-observations of the danube data set.

Method-of-moments estimators can be used to fit the Gumbel–Hougaard, the Plackett or the normal copula family to these bivariate data:

```
1 > summary(fitCopula(gumbelCopula(), data = U, method = "itau"))
```

```
Call: fitCopula(copula, data = data, method = "itau")
Fit based on "inversion of Kendall's tau" and 659 2-dimensional observations.
Gumbel copula, dim. d = 2
    Estimate Std. Error
alpha     2.215      0.091
```

```
1 > summary(fitCopula(plackettCopula(), data = U, method = "itau"))
```

```
Call: fitCopula(copula, data = data, method = "itau")
Fit based on "inversion of Kendall's tau" and 659 2-dimensional observations.
plackettCopula copula, dim. d = 2
    Estimate Std. Error
alpha     15.21      1.729
```

```
1 > summary(fitCopula(normalCopula(), data = U, method = "itau"))
```

```
Call: fitCopula(copula, data = data, method = "itau")
Fit based on "inversion of Kendall's tau" and 659 2-dimensional observations.
Normal copula, dim. d = 2
    Estimate Std. Error
rho.1    0.7588      0.019
```



4.1.2.2 Maximum pseudo-likelihood estimator

- If the margins F_1, \dots, F_d were known, the maximum likelihood estimator of θ_0 would be given by

$$\boldsymbol{\theta}_n = \operatorname{argsup}_{\boldsymbol{\theta} \in \Theta} \sum_{i=1}^n \log c_{\boldsymbol{\theta}}(\mathbf{U}_i)$$

in terms of the iid sample $\mathbf{U}_1, \dots, \mathbf{U}_n$ defined in (4.2).

- Since the margins are unknown, it seems natural to replace $\mathbf{U}_1, \dots, \mathbf{U}_n$ by the pseudo-observations $\mathbf{U}_{i,n} = (F_{n,1}(X_{i1}), \dots, F_{n,d}(X_{id})), i \in \{1, \dots, n\}$, see (4.8), thereby leading to the maximum pseudo-likelihood estimator (MPLE)

$$\boldsymbol{\theta}_n = \operatorname{argsup}_{\boldsymbol{\theta} \in \Theta} \sum_{i=1}^n \log c_{\boldsymbol{\theta}}(\mathbf{U}_{i,n}). \quad (4.10)$$

- For copula families whose parameters can be estimated by method-of-moments estimators, the MPLE was observed to be more efficient.

- Also, the range of applicability of the MPLE is greater than the one of the method-of-moments estimators, especially for multiparameter copulas. As long as the density c_θ can be evaluated, one can attempt to maximize the pseudo-likelihood.
- The (numerical) optimization can typically be made more efficient with initial values for θ_0 (for example, via method-of-moments estimators). However, it can become unstable when d is large, the number p of parameters is large or the sample size n is small.
- Standard errors for the MPLE can be computed as explained in Genest et al. (1995); see also Kojadinovic and Yan (2010a, Section 3). Their estimation requires the computation of certain partial derivatives of the (log) density. For those families in copula for which these partial derivatives are not implemented, numerical differentiation based on the R package `numDeriv` of Gilbert and Varadhan (2016) is used (and a warning message is displayed).

Example 4.1.7 (Estimation of copula parameters via the MPLE)

Assume that \mathcal{C} is the Frank copula family. The MPLE for θ_0 for any meta-Frank distribution with copula parameter $\theta_0 = 3$ and continuous margins can be computed as follows:

```
1 > ## The "unknown" copula (a 2-dim. Frank copula with parameter 3)
2 > fc <- frankCopula(3)
3 > ## Generate the "observed" sample
4 > set.seed(271)
5 > U <- rCopula(1000, fc)
6 > ## Compute the MPLE and its standard error
7 > summary(fitCopula(frankCopula(), data = pobs(U), method = "mpl"))
```

```
Call: fitCopula(copula, data = data, method = "mpl")
Fit based on "maximum pseudo-likelihood" and 1000 2-dimensional observations.
Frank copula, dim. d = 2
    Estimate Std. Error
alpha     2.907      0.2
The maximized loglikelihood is 101.5
Optimization converged
Number of loglikelihood evaluations:
function gradient
        4          4
```

- Similar to method-of-moments estimators, for the standard errors returned by `summary(fitCopula(, method = "mpl"))` to be consistently estimated, `fitCopula()` needs to be called on pseudo-observations.
- As explained in Example 4.1.2, the default optimization method is determined by `optimMeth()`. For the bivariate Frank family, we have:

```
1 > optimMeth(frankCopula(), method = "mpl", dim = 2)
```

```
[1] "L-BFGS-B"
```



Example 4.1.8 (Application of the estimation via the MPLE)

- As a second example of MPLE, consider again the trivariate daily log-return data set (Intel, Microsoft, General Electric) introduced in Example 4.1.3.
- Fit two trivariate copula families to these data: the (non-homogeneous) normal and *t* families.

- It is necessary to specify that the correlation matrices P of C_P^n and $C_{P,\nu}^t$ should be considered as parametrized by $\binom{d}{2}$ elements with $d = 3$ (instead of just one correlation parameter as in the homogeneous case). This can be done by changing the default value of the argument `dispstr` of `normalCopula()` and `tCopula()` from "ex" (for "exchangeable") to "un" (for "unstructured"):

```

1 > U <- pobs(rdj[,2:4]) # compute the pseudo-observations
2 > ## MPLE for the normal copula
3 > summary(fitCopula(normalCopula(dim = 3, dispstr = "un"), data = U))

```

```

Call: fitCopula(copula, data = data)
Fit based on "maximum pseudo-likelihood" and 1262 3-dimensional observations.
Normal copula, dim. d = 3
    Estimate Std. Error
rho.1    0.5781    0.016
rho.2    0.3400    0.020
rho.3    0.4017    0.018
The maximized loglikelihood is 375.7
Optimization converged
Number of loglikelihood evaluations:
function gradient
      29          6

```

```
1 > ## MPLE for the t copula  
2 > summary(fitCopula(tCopula(dim = 3, dispstr = "un"), data = U))
```

```
Call: fitCopula(copula, data = data)  
Fit based on "maximum pseudo-likelihood" and 1262 3-dimensional observations.  
t-copula, dim. d = 3  
    Estimate Std. Error  
rho.1    0.5877    0.020  
rho.2    0.3593    0.026  
rho.3    0.4225    0.025  
df       6.5025    NA  
The maximized loglikelihood is 419.3  
Optimization converged  
Number of loglikelihood evaluations:  
function gradient  
      31        10
```

- Note that `summary(fitCopula(, method = "mpl"))` ("mpl" being the default value of the argument `method`) does not return a standard error for the estimate of the degrees of freedom of a *t* copula.
- The standard errors of the correlation parameters are estimated as if the

underlying t copula had its degree of freedom parameter fixed (to its estimated value); see Section 4.1.5 below.

- As a consequence, the computed standard errors underestimate, on average, the true unknown standard errors (simulations indicate that the bias might not be of practical importance, though).
- These shortcomings are, among other things, due to the fact that the t copula cannot be evaluated for non-integer degrees of freedom for the moment, as already mentioned in Example 2.3.3.



4.1.3 Estimators of elliptical copula parameters

- For large d , MPLE for elliptical copulas becomes prohibitive and numerically unstable because of the number of parameters (which is $\binom{d}{2}$ for the correlation matrix alone).

- Method-of-moments estimators based on Spearman's rho or Kendall's tau remain a feasible alternative. First compute the d by d matrix of sample Spearman's rhos (respectively, Kendall's taus) and then apply $g_{\rho_s}^{-1}$ (respectively, g_τ^{-1}) element-wise to obtain an estimate of P .
- To guarantee that the estimate of P is indeed a proper correlation matrix, the method of Higham (2002) is applied by default.

Example 4.1.9 (Estimation of normal copula parameters via method-of-moments)

Consider pseudo-observations computed from log-returns of all $d = 20$ constituents of the Swiss Market Index (SMI) from 2011-09-09 to 2012-03-28.

```
1 > data(SMI.12) # load the SMI constituent data
2 > library(qrmtools)
3 > X <- returns(SMI.12) # compute log-returns
4 > U <- pobs(X) # compute pseudo-observations
5 > d <- ncol(U) # 20 dimensions
```

Fit a normal copula C_P^n to these pseudo-observations using method-of-

moments estimators based on Spearman's rho and Kendall's tau:

```
1 > f.irho <- fitCopula(normalCopula(dim = d, dispstr = "un"), data = U,  
2                         method = "irho")  
3 > f.itau <- fitCopula(normalCopula(dim = d, dispstr = "un"), data = U,  
4                         method = "itau")
```

$f.\text{irho}@\text{estimate}$ and $f.\text{itau}@\text{estimate}$ are vectors of $\binom{d}{2}$ elements.
They can be converted to correlation matrices using `p2P()`:

```
1 > P.irho <- p2P(f.irho@estimate, d = d)  
2 > P.itau <- p2P(f.itau@estimate, d = d)
```

Notice that the `inverse operation` is available via the function `P2p()`.

The argument `posDef` of `fitCopula()` is TRUE by default when called on an object representing an elliptical copula. This results in internally applying `nearPD()` of the R package `Matrix`, which computes the nearest positive definite matrix to the initial correlation matrix estimate and guarantees a proper correlation matrix; see Higham (2002). □

- Because of the degrees of freedom parameter ν , this approach is **not directly applicable to the t family**.
- In this case one can **first** use pairwise inversion of Kendall's tau for estimating the $\binom{d}{2}$ correlation parameters in P and **then** maximum pseudo-likelihood-like estimation based on the estimated P and the available pseudo-observations **to estimate ν** .

Example 4.1.10 (Estimation of t copula parameters using the method of Mashal and Zeevi)

Fit a t family using the method of Mashal and Zeevi (2002) **to the pseudo-observations of Example 4.1.9 (SMI data)**:

```
1 > fit <- fitCopula(tCopula(dim = d, dispstr = "un"), data = U,
2                      method = "itau.mpl")
```

The estimation of the standard errors is not implemented yet. □

- As mentioned in Section 2.6, using sample (linear) correlation is truly meaningful only when the observations arise from an elliptical distribution.
- In case of the multivariate t distribution, for instance, the sample correlation matrix is a natural estimator of the parameter P of the underlying t copula.
- The following example compares this estimator of P with the method-of-moments estimator based on inversion of Kendall's tau through simulations.

Example 4.1.11 (Linear correlation vs Kendall's tau for estimating t distributions)

We write an auxiliary function to draw B samples of size n from a bivariate t distribution (fixed ν) with correlation parameter ρ and then compute for each of the B samples, both estimates of ρ :

```
1 > #' @title Compute (repeated) parameter estimates for the bivariate t
2 > #' @param nu degrees of freedom parameter
3 > #' @param rho correlation parameter
4 > #' @param B number of replications
5 > #' @param n sample size
6 > #' @return (B, 2)-matrix containing the B estimates computed
7 > #'           via Pearson's rho and Kendall's tau
8 > estimates <- function(nu, rho, B, n)
9 {
10   ## Generate data (B-list of (n, 2)-matrices)
11   tc <- tCopula(rho, df = nu) # define the corresponding t copula
12   X <- lapply(1:B, function(i) qt(rCopula(n, copula = tc), df = nu))
13   ## For each of the B data sets, estimate the correlation parameter of the
14   ## t distribution via Pearson's rho and Kendall's tau.
15   pea <- vapply(X, function(x) cor(x[,1], x[,2]), numeric(1))
16   ken <- iTau(tCopula(df = nu),
17               tau = sapply(X, function(x) cor(x, method = "kendall")[2,1]))
18   ## Return
19   cbind(Pearson = pea, Kendall = ken) # (B, 2)-matrix
20 }
```

We use this function to generate $B = 3000$ samples of size $n = 90$ from a bivariate t distribution with $\rho = 0.5$ and $\nu = 3$:

```
1 > set.seed(271) # for reproducibility
2 > nu <- 3 # degrees of freedom
3 > rho <- 0.5 # correlation parameter
4 > r <- estimates(nu, rho = rho, B = 3000, n = 90) # (B, 2)-matrix
5 > varP <- var(r[, "Pearson"]) # variance of sample linear correlation
6 > varK <- var(r[, "Kendall"]) # variance of inverting Kendall's tau
7 > VRF <- varP / varK # variance reduction factor
8 > PIM <- (varP - varK) / varP * 100 # % improvement
9 > boxplot(r, names = c("Sample linear correlation", "Inverting Kendall's tau"),
10           ylab = substitute("Estimates of `~rho~` of a"
11                           ~t[nu.]~"distribution with"~rho==rho.,
12                           list(nu. = nu, rho. = rho)))
13 > mtext(substitute("VRF (% improvement):"~~v~"(*i*%)",
14                     list(v = round(VRF, 2), i = round(PIM))),
15                     side = 4, line = 1, adj = 0, las = 0)
```

By simulation, we can also investigate this for as a function of ν and p :

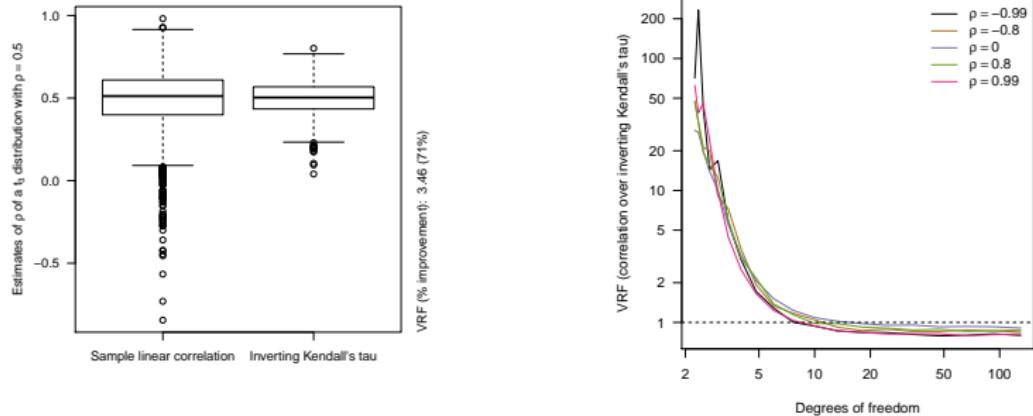


Figure 4.4 Box plots of sample linear correlation and method-of-moments estimates of ρ based on the inversion of Kendall's tau for $B = 3000$ samples of size $n = 90$ from a bivariate t distribution with $\nu = 3$ degrees of freedom and correlation parameter $\rho = 0.5$ (left); note the smaller variance of the estimator based on the inversion of Kendall's tau in this case. Estimated variance-reduction factor (VRF) as a function of the degrees of freedom ν and the correlation parameter ρ (right).

- Inversion of Kendall's tau has a **smaller variance if $\nu > 2$ is small.**
- For moderate ν , sample correlation has the smallest variance when $|\rho|$ is large.
- **For large ν , sample correlation appears to be better.** □

4.1.4 Other semi-parametric estimators

- Tsukahara (2005) investigates rank approximate Z -estimators and minimum-distance estimators. In a simulation study, both led to a higher estimated mean squared error than the MPLE, though.
- A version of the MPLE where the unknown marginal densities are approximated by linear combinations of finite-dimensional basis functions with increasing complexity, called sieves, is investigated in Chen et al. (2006). The authors show that the resulting estimator of θ_0 is asymptotically semi-parametrically efficient provided additional smoothness conditions are satisfied.
- The study of the finite-sample performance of the method (for $n = 400$) reveals that this approach performs significantly better than the standard MPLE when one marginal df is known. This advantage does not seem to hold anymore when all marginals are unknown.

4.1.5 Estimation of copula models with partly fixed parameters

- We might want to fix some components of the parameter vector θ .
- `fixParam()` (controls which component should be fixed when creating an R object representing a copula) and `fixedParam()` (modify an existing R object in terms of which parameters are fixed).

Example 4.1.12 (Estimation of elliptical copulas with partly fixed parameters)

Consider estimating the parameters of a trivariate normal copula family whose first parameter is fixed to 0.6:

```
1 > ## The "unknown" copula (a 3-dim. normal copula)
2 > nc <- normalCopula(param = c(0.6, 0.3, 0.2), dim = 3, dispstr = "un")
3 > ## Generate the "observed" sample and compute corresponding pobs
4 > set.seed(819)
5 > U <- pobs(rCopula(1000, nc))
6 > ## A trivariate normal copula whose first parameter is fixed to 0.6
7 > (ncf <- normalCopula(param = fixParam(c(0.6, NA_real_, NA_real_),
8 >                               c(TRUE, FALSE, FALSE)),
9 >                               dim = 3, dispstr = "un"))
```

```
Normal copula, dim. d = 3
Dimension: 3
Parameters (partly fixed, see ':='):
  rho.1 := 0.6
  rho.2 = NA
  rho.3 = NA
dispstr: un
```

```
1 > fitCopula(ncf, data = U) # MPLE
```

```
Call: fitCopula(copula, data = data)
Fit based on "maximum pseudo-likelihood" and 1000 3-dimensional observations.
Copula: normalCopula
  rho.2 rho.3
0.2882 0.1834
The maximized loglikelihood is 283.3
Optimization converged
```

```
1 > fitCopula(ncf, data = U, method = "itau")
```

```
Call: fitCopula(copula, data = data, method = "itau")
Fit based on "inversion of Kendall's tau" and 1000 3-dimensional observations.
Copula: normalCopula
  rho.2 rho.3
0.2861 0.1952
```

```
1 > fitCopula(ncf, data = U, method = "irho")
```

```
Call: fitCopula(copula, data = data, method = "irho")
Fit based on "inversion of Spearman's rho" and 1000 3-dimensional observations.
Copula: normalCopula
rho.2 rho.3
0.2863 0.1948
```

Alternatively, fix the parameter values of an existing copula object using the function `fixedParam()`:

```
1 > fixedParam(nc) <- c(TRUE, FALSE, FALSE)
2 > nc
```

```
Normal copula, dim. d = 3
Dimension: 3
Parameters (partly fixed, see ':='):
  rho.1 := 0.6
  rho.2 = 0.3
  rho.3 = 0.2
dispstr: un
```

One can now simply call `fitCopula(nc,)` to fit the normal copula family with first correlation parameter fixed to 0.6.

Similarly for the t copula family, this time with two fixed correlation parameters:

```
1 > ## The "unknown" copula is a 3-dim. t copula (with 4 d.o.f. by default)
2 > tc <- tCopula(param = c(0.6, 0.3, 0.2), dim = 3, dispstr = "un")
3 > ## Generate the "observed" sample and compute corresponding pobs
4 > set.seed(314)
5 > U <- pobs(rCopula(1000, tc))
6 > ## A trivariate t copula whose first two parameters are fixed to 0.6 and 0.3
7 > (tcf <- tCopula(param = fixParam(c(0.6, 0.3, NA_real_),
8 >                      c(TRUE, TRUE, FALSE)),
9 >                      dim = 3, dispstr = "un"))
```

```
t-copula, dim. d = 3
Dimension: 3
Parameters (partly fixed, see ':='):
  rho.1  := 0.6
  rho.2  := 0.3
  rho.3  = NA
  df     = 4.0
dispstr: un
```

```
1 > fitCopula(tcf, data = U) # MPLE
```

```
Call: fitCopula(copula, data = data)
```

```
Fit based on "maximum pseudo-likelihood" and 1000 3-dimensional observations.  
Copula: tCopula  
rho.3      df  
0.2208 4.2502  
The maximized loglikelihood is 366.1  
Optimization converged
```

```
1 > fitCopula(tcf, data = U, method = "itau.mpl")
```

```
Call: fitCopula(copula, data = data, method = "itau.mpl")  
Fit based on "itau for dispersion matrix P and maximum likelihood for df" and 1000 3-  
dimensional observations.  
Copula: tCopula  
rho.3      df  
0.246 4.257  
The maximized loglikelihood is 365.6  
Optimization converged
```

To reduce the complexity of the inference for the t family, it can be useful to keep the [degrees of freedom fixed](#):

```
1 > ## A trivariate t copula whose first correlation parameter is fixed to 0.6  
2 > ## and whose number of degrees of freedom is fixed to 4 (default value)  
3 > (tcf2 <- tCopula(param = fixParam(c(0.6, NA_real_, NA_real_),  
4                               c(TRUE, FALSE, FALSE)),
```

```
5           dim = 3, dispstr = "un", df.fixed = TRUE))
```

```
t-copula, dim. d = 3
Dimension: 3
Parameters (partly fixed, see ':='):
  rho.1  := 0.6
  rho.2  = NA
  rho.3  = NA
  df     := 4.0
dispstr: un
```

```
1 > fitCopula(tcf2, data = U) # MPLE
```

```
Call: fitCopula(copula, data = data)
Fit based on "maximum pseudo-likelihood" and 1000 3-dimensional observations.
Copula: tCopula
  rho.2 rho.3
0.3241 0.2337
The maximized loglikelihood is 366.2
Optimization converged
```

```
1 > fitCopula(tcf2, data = U, method = "itau")
```

```
Call: fitCopula(copula, data = data, method = "itau")
Fit based on "inversion of Kendall's tau" and 1000 3-dimensional observations.
```

```
Copula: tCopula  
rho.2 rho.3  
0.3393 0.2460
```



Example 4.1.13 (Estimation of Khoudraji–Clayton copulas with partly fixed parameters)

Consider fitting the three-parameter Khoudraji–Clayton copula $\text{kho}_s(\Pi, C_\theta^C)$:

```
1 > ## The "unknown" copula (a 2-dim. Khoudraji-Clayton copula)  
2 > kc <- khoudrajiCopula(copula2 = claytonCopula(6), shapes = c(0.4, 1))  
3 > set.seed(1307)  
4 > U <- pobs(rCopula(1000, kc))  
5 > ## The default optimization method for fitting bivariate copulas  
6 > ## constructed with Khoudraji's device by MPLE  
7 > optimMeth(khoudrajiCopula(), method = "mpl", dim = 2)
```

```
[1] "BFGS"
```

Fitting $\text{kho}_s(\Pi, C_\theta^C)$ using the BFGS optimization method fails:

```
1 > try(fitCopula(khoudrajiCopula(copula2 = claytonCopula()),
2                  start = c(1.1, 0.5, 0.5), data = U))
```

```
Error in optim(start, logL, lower = lower, upper = upper, method = optim.method, :
non-finite finite-difference value [3]
```

The same task based on the more robust Nelder-Mead method:

```
1 > fitCopula(khoudrajiCopula(copula2 = claytonCopula()),
2                  start = c(1.1, 0.5, 0.5), data = U,
3                  optim.method = "Nelder-Mead")
```

```
Call: fitCopula(copula, data = data, start = ..1, optim.method = "Nelder-Mead")
Fit based on "maximum pseudo-likelihood" and 1000 2-dimensional observations.
Copula: khoudrajiExplicitCopula
c2.alpha    shape1    shape2
  5.9257    0.3929    1.0000
The maximized loglikelihood is 166.5
Optimization converged
```

Instead, one could consider the two-parameter $\text{kho}_{(s_1, 1)}(\Pi, C_\theta^C)$, where fitting works even with BFGS:

```
1 > kcf <- khoudrajiCopula(copula2 = claytonCopula(),
2                               shapes = fixParam(c(NA_real_, 1),
3                                                 c(FALSE, TRUE)))
4 > fitCopula(kcf, start = c(1.1, 0.5), data = U)
```

```
Call: fitCopula(copula, data = data, start = ..1)
Fit based on "maximum pseudo-likelihood" and 1000 2-dimensional observations.
Copula: khoudrajiExplicitCopula
c2.alpha    shape1
 5.9249     0.3929
The maximized loglikelihood is 166.5
Optimization converged
```



4.2 Non-parametric estimation of the copula

- We consider the simplest, most studied non-parametric estimator of C , the so-called empirical copula.
- We also briefly mention a class of non-parametric estimators if C is an extreme-value copula.

4.2.1 The empirical copula

- In the same way that the empirical df is a natural non-parametric estimator of a df, a sensible non-parametric estimator of C is

$$C_n(\mathbf{u}) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}(U_{i,n} \leq \mathbf{u}) = \frac{1}{n} \sum_{i=1}^n \prod_{j=1}^d \mathbf{1}(U_{ij,n} \leq u_j), \quad \mathbf{u} \in [0, 1]^d,$$

where $U_{1,n}, \dots, U_{n,n}$ are the **pseudo-observations** defined in (4.8).

- The quantity C_n , which is merely the empirical df of the pseudo-observations, is referred to as the **empirical copula** of $\mathbf{X}_1, \dots, \mathbf{X}_n$.
- It is a consistent estimator of C whose asymptotics follow from those of the so-called **empirical copula process**

$$\sqrt{n}(C_n(\mathbf{u}) - C(\mathbf{u})), \quad \mathbf{u} \in [0, 1]^d. \quad (4.11)$$

- The empirical copula process arises, in particular, as a key theoretical tool for showing the asymptotic validity of many inference procedures on C .

Example 4.2.1 (Non-parametric estimation by the empirical copula)

- Assume that C is the Clayton copula C_3^C with parameter 3.
- Non-parametric estimation of C by the empirical copula of a random sample of size n from any meta- C_3^C distribution with continuous margins:

```
1 > ## The "unknown" copula (a 3-dim. Clayton copula with parameter 3)
2 > d <- 3
3 > cc <- claytonCopula(3, dim = d)
4 > ## Generate a sample from the copula, which will be transformed
5 > ## to pseudo-observations in 'C.n()'
6 > n <- 1000
7 > set.seed(65)
8 > U <- rCopula(n, copula = cc)
9 > ## Generate random points where to evaluate the empirical copula
10 > v <- matrix(runif(n * d), nrow = n, ncol = d)
11 > ec <- C.n(v, X = U)
12 > ## Compare with the true copula; increase n to decrease the error
13 > true <- pCopula(v, copula = cc)
14 > round(mean(abs(true - ec) / true) * 100, 2) # mean relative error (in %)
```

```
[1] 1.55
```



- As a particular multivariate empirical df, the empirical copula often exhibits a large bias when the sample size is small.
- A class of smooth non-parametric estimators of C are so-called empirical Bernstein copulas.
- One member of interest is the *empirical beta copula* of Segers et al. (2017) which is the empirical Bernstein copula obtained by setting the degrees of all Bernstein polynomials equal to the sample size.
- The empirical beta copula of X_1, \dots, X_n can be computed as

$$C_n^\beta(\mathbf{u}) = \frac{1}{n} \sum_{i=1}^n \prod_{j=1}^d F_{n,R_{ij}}(u_j), \quad \mathbf{u} \in [0, 1]^d,$$

where, for any $r \in \{1, \dots, n\}$, $F_{n,r}$ denotes the df of the beta distribution with parameters r and $n+1-r$, respectively, and where, for any $j \in \{1, \dots, d\}$, R_{ij} is the rank of X_{ij} among X_{1j}, \dots, X_{nj} .

- The smoothness of C_n^β is a consequence of the replacement of the indicator function by a product of particular beta dfs.
- Unlike the empirical copula C_n , the empirical beta copula C_n^β has standard uniform univariate margins (as long as there are no ties in the components samples of $\mathbf{X}_1, \dots, \mathbf{X}_n$) and is therefore a genuine copula in that case.
- An even simpler smooth version of the empirical copula is the so-called *empirical checkerboard copula* of $\mathbf{X}_1, \dots, \mathbf{X}_n$ defined by

$$C_n^\#(\mathbf{u}) = \frac{1}{n} \sum_{i=1}^n \prod_{j=1}^d \min\{\max\{nu_j - R_{ij} + 1, 0\}, 1\}, \quad \mathbf{u} \in [0, 1]^d.$$

Being a multilinear extension of (an asymptotically equivalent version of) the empirical copula C_n , it is again a genuine copula as long as there are no ties in the components samples of $\mathbf{X}_1, \dots, \mathbf{X}_n$.

Example 4.2.2 (The empirical beta and checkerboard copulas)

- Empirical beta and checkerboard copulas can be evaluated with `C.n()` with argument chosen as `smoothing = "beta"` and `smoothing = "checkerboard"`, respectively.
- We now illustrate the improved small-sample behavior of these smooth non-parametric estimators compared to the empirical copula.
- Generate a sample of size n from an “unknown” Gumbel–Hougaard copula, evaluate the three empirical copulas at random points and return the corresponding mean relative errors as percentages:

```
1 > gc <- gumbelCopula(4, dim = 3) # the 'unknown' copula
2 > #' @title Mean relative error in % (for empirical copula, empirical beta
3 > #'          copula and empirical checkerboard copula)
4 > #' @param n sample size
5 > #' @param cop copula object
6 > #' @return mean relative errors in %
7 > compareEmpCops <- function(n, cop)
8 {
9     d <- dim(cop)
10    U <- rCopula(n, copula = cop) # a sample from the true copula
```

```

11     v <- matrix(runif(n * d), nrow = n, ncol = d) # random evaluation points
12     ec   <- C.n(v, X = U) # the empirical copula values
13     beta <- C.n(v, X = U, smoothing = "beta") # the emp. beta cop. values
14     check <- C.n(v, X = U, smoothing = "checkerboard") # emp. check. cop. val
15     true <- pCopula(v, copula = cop) # the true copula values
16     c(ec   = mean(abs(true - ec) / true),
17          beta = mean(abs(true - beta) / true),
18          check = mean(abs(true - check) / true)) * 100 # mean rel. error in %
19 }

```

Execute the above function 100 times and return the average of the errors:

```

1 > set.seed(2013)
2 > round(rowMeans(replicate(100, compareEmpCops(30, cop = gc))), 2)

```

	ec	beta	check
20.68	6.52	7.52	

- Hence, the empirical beta and checkerboard copulas have a smaller estimated mean relative error than the empirical copula.
- This advantage progressively vanishes as the sample size increases:

```
1 > set.seed(2013)
2 > round(rowMeans(replicate(100, compareEmpCops(300, cop = gc))), 2)
```

```
ec  beta check
4.40 2.07 2.45
```

- Let's check that the two smooth estimators indeed have $U(0, 1)$ margins:

```
1 > set.seed(2008)
2 > U <- rCopula(30, copula = gc) # a sample from the true copula
3 > m <- 100 # number of evaluation points
4 > v <- runif(m) # random points where to evaluate the margins of the estimators
5 > w1 <- cbind(v, matrix(1, nrow = m, ncol = dim(gc) - 1)) # eval. pts. margin 1
6 > w2 <- cbind(matrix(1, nrow = m, ncol = dim(gc) - 1), v) # eval. pts. margin 2
7 > stopifnot(all.equal(C.n(w1, X = U, smoothing = "beta"), v)) # check
8 > stopifnot(all.equal(C.n(w2, X = U, smoothing = "beta"), v)) # check
9 > stopifnot(all.equal(C.n(w1, X = U, smoothing = "checkerboard"), v)) # check
10 > stopifnot(all.equal(C.n(w2, X = U, smoothing = "checkerboard"), v)) # check
```



4.2.2 Under extreme-value dependence

- Assume C is of extreme-value type; for example, if X_1, \dots, X_n are componentwise block maxima.
- C can be expressed in terms of the Pickands dependence function A .
- Given a non-parametric estimator A_n of A , a sensible non-parametric estimator of C can then be obtained by plugging A_n into (3.2).
- See the functions `An.biv` and `An`.

5 Graphical diagnostics, tests and model selection

5.1 Basic graphical diagnostics

5.2 Hypothesis tests

5.3 Model selection

- We assume to have at hand a random sample $\mathbf{X}_1, \dots, \mathbf{X}_n$ from a df H with **continuous** univariate margins F_1, \dots, F_d .
- By **Sklar's Theorem**, H can be expressed as

$$H(\mathbf{x}) = C(F_1(x_1), \dots, F_d(x_d)), \quad \mathbf{x} \in \mathbb{R}^d, \quad (5.1)$$

- In the previous chapter, we discussed the estimation of the copula under the **realistic assumption** that F_1, \dots, F_d are **unknown** and that C belongs to an absolutely continuous parametric family of copulas $\mathcal{C} = \{C_\theta : \theta \in \Theta\}$, where Θ , the parameter space, is a subset of \mathbb{R}^p for some integer $p \geq 1$.
- The aim of this chapter is to present some **graphical** and **formal statistical procedures** that can be used to **guide the choice** of the hypothesized parametric copula family \mathcal{C} .
- Should the margins F_1, \dots, F_d be known, we could form the sample $\mathbf{U}_1, \dots, \mathbf{U}_n$, where $\mathbf{U}_i = (F_1(X_{i1}), \dots, F_d(X_{id}))$, and carry out inference on C using **standard statistical procedures** for iid observations.

- As the sample $\mathbf{U}_1, \dots, \mathbf{U}_n$ is **unobservable**, a natural path is to form parametric or non-parametric pseudo-observations instead, as defined by (4.5) or (4.8), respectively.
- To avoid biases arising from potential **misspecifications** of the margins, many authors advocate to base the inference on C on the latter, which amounts to forming the multivariate scaled ranks (4.9) as explained in the previous chapter.
- We adopt the same perspective in this chapter, which implies that all the presented procedures are **margin-free**.
- We shall first briefly discuss some **basic graphical diagnostics** to guide the choice of \mathcal{C} .
- Several **formal statistical tests** will be presented next.
- The last section of the chapter will be devoted to the key issue of **model selection**, that is, copula family selection.

5.1 Basic graphical diagnostics

- Let us assume for simplicity that $d = 2$; if $d > 2$ but is not too large, the suggested diagnostics can be performed on all $\binom{d}{2}$ bivariate margins.
- One of the first steps when analyzing $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n$ consists of producing a **scatter plot of the pseudo-observations** defined in (4.8).
- Such a scatter plot could for instance confirm that the underlying copula C is indeed different from the independence copula Π , thereby justifying its modeling by a parametric family.
- The scatter plot of the pseudo-observations can for instance also suggest that the underlying copula is radially symmetric, that is, that $C = \bar{C}$, or that it is not exchangeable, that is, that $C(u_1, u_2) = C(u_2, u_1)$ does not hold for all $u_1, u_2 \in [0, 1]$.
- To assess the presence of **tail dependence**, the pseudo-observations (4.8) can be transformed to **normal scores** first, by applying the quantile

function of the standard normal to the component samples of the pseudo-observations.

- The rationale behind this approach is that normal scores are **approximately marginally standard normal** and that the resulting scatter plot could thus highlight departures from bivariate normality.
- Indeed, a scatter plot of bivariate normal observations should look “**elliptically contoured**”.
- **Heavier tails** (having more probability mass than the tails of normal distributions) in the scatter plot of normal scores suggest **tail dependence**; see, for example, Joe (2014, Section 1.4) for more details.
- In addition, the non-exchangeability or radial symmetry of C might be easier to detect than from the scatter plot of pseudo-observations.

Example 5.1.1 (Pseudo-observations and normal scores)

The **pseudo-observations** and **normal scores** of the danube data set can be computed as follows (see Figure 5.1):

```
1 > data(danube, package = "lcopula")
2 > U <- as.matrix(danube)
3 > plot(U, xlab = "Donau", ylab = "Inn")
4 > plot(qnorm(U), xlab = "Donau", ylab = "Inn")
```

- Both scatter plots seem to show a larger number of observations above the main diagonal, thereby suggesting a **slight departure from exchangeability**.
- The **lack of symmetry** with respect to the point $(1/2, 1/2)$ also suggests that C is **not radially symmetric**.
- Finally, both the scatter plots of pseudo-observations and normal scores suggest the presence of (at least) **upper tail dependence**.



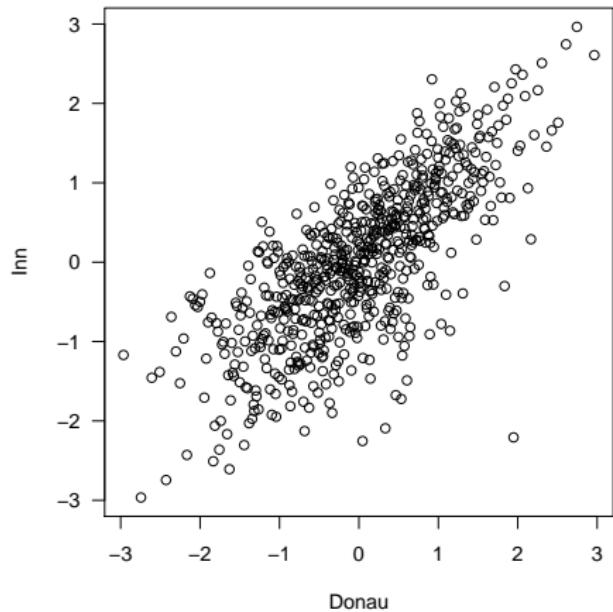
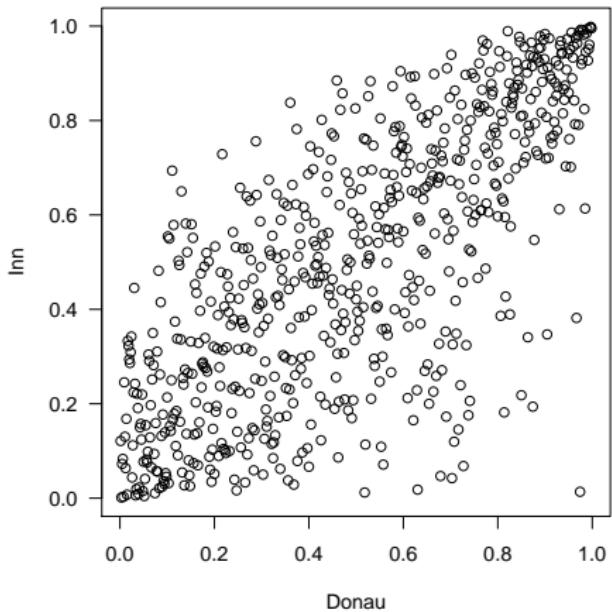


Figure 5.1 Pseudo-observations (left) and normal scores (right) of the danube data set.

- The previous basic graphical diagnostics might suggest to mainly focus on certain parametric copula families for \mathcal{C} .
- For each such family, C could be estimated using one of the approaches described in the previous chapter.
- To assess which estimated copula fits best, some authors like to compare **contours plots** of the **parametric estimates** with those of the **empirical copula** of the data.

Example 5.1.2 (Comparing (non-)parametric estimates of the copula)

- The scatter plots produced in the previous example suggest for instance to consider the **Joe** or the **Gumbel–Hougaard** family as possible choices for \mathcal{C} , as both of these families exhibit **upper tail dependence**.
- Figure 5.2, produced with the following code (based on the package `latticeExtra` of Sarkar (2016) for overlaying plots), compares the corresponding contours with those of the empirical copula:

```
1 > ## Fit a Gumbel-Hougaard copula and compute the contours
2 > fg <- fitCopula(gumbelCopula(), data = U)
3 > cpG <- contourplot2(fg@copula, FUN = pCopula, region = FALSE,
4   key = list(corner = c(0.04, 0.04),
5     lines = list(col = 1:2, lwd = 2),
6     text = list(c("Fitted Gumbel-Hougaard copula",
7       "Empirical copula")))))
8 > ## Fit a Joe copula and compute the contours
9 > fj <- fitCopula(joeCopula(), data = U)
10 > cpJ <- contourplot2(fj@copula, FUN = pCopula, region = FALSE,
11   key = list(corner = c(0.04, 0.04),
12     lines = list(col = 1:2, lwd = 2),
13     text = list(c("Fitted Joe copula",
14       "Empirical copula")))))
15 > ## Compute the contours of the empirical copula
16 > u <- seq(0, 1, length.out = 16)
17 > grid <- as.matrix(expand.grid(u1 = u, u2 = u))
18 > val <- cbind(grid, z = C.n(grid, X = U))
19 > cpCn <- contourplot2(val, region = FALSE, labels = FALSE, col = 2)
20 > ## Plots (lattice objects)
21 > library(latticeExtra)
22 > cpG + cpCn
23 > cpJ + cpCn
```

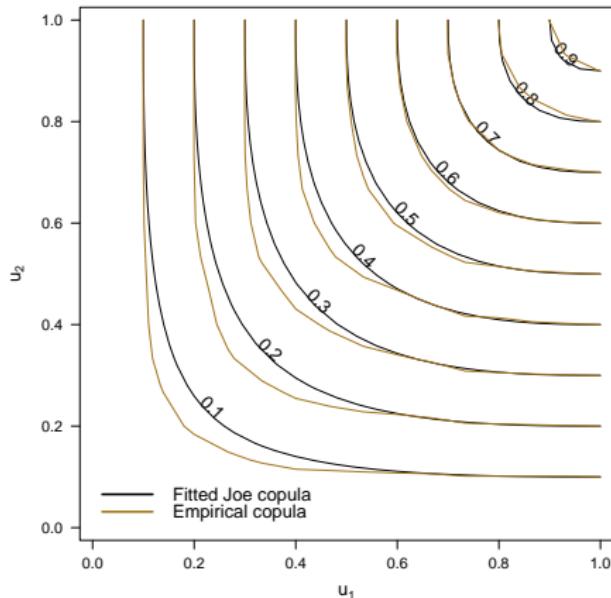
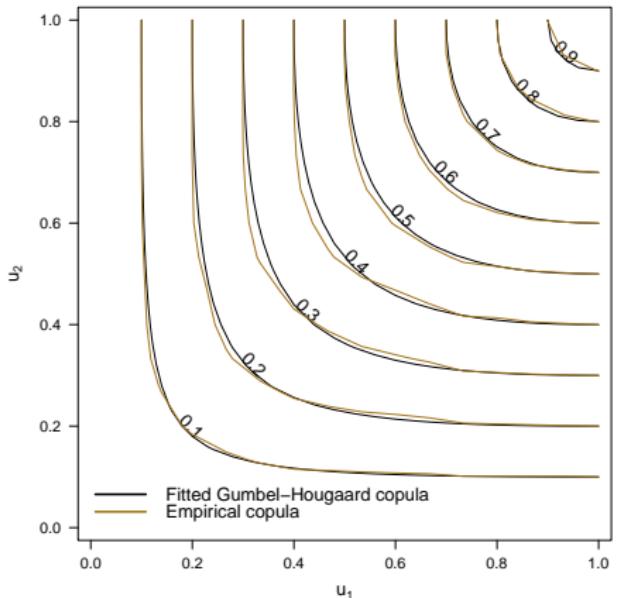


Figure 5.2 Contour plots of the fitted Gumbel–Hougaard copula (left) and Joe copula (right) overlaid by those of the empirical copula for the danube data set.

- The contour plots in Figure 5.2 suggest that the Gumbel–Hougaard family **fits better** than the Joe family.

- Because of the upper tail dependence and the possible slight non-exchangeability of C seeming to appear in Figure 5.1, it is also natural to consider the [Khoudraji–Gumbel–Hougaard](#) family $\text{kho}_s(\Pi, C_\theta^{\text{GH}})$ constructed as explained in Section 3.4.2 as candidate family.
- The fitting of families constructed using Khoudraji's device can be **numerically challenging** as already discussed in Example 4.1.13, which is why the more robust [Nelder–Mead](#) optimization routine is used in the forthcoming code.
- Starting values for the parameters also need to be provided.

```

1 > (fk <- fitCopula(khoudrajiCopula(copula2 = gumbelCopula()), data = U,
2                           start = c(1.1, 0.5, 0.5), optim.method = "Nelder-Mead"))

```

```

Call: fitCopula(copula, data = data, start = ..1, optim.method = "Nelder-Mead")
Fit based on "maximum pseudo-likelihood" and 659 2-dimensional observations.
Copula: khoudrajiExplicitCopula
c2.alpha    shape1    shape2
 2.2661    0.9201   1.0000
The maximized loglikelihood is 281.9
Optimization converged

```

- As we have already seen in the previous chapter, one way to make the optimization problem easier is to decrease the number of parameters by keeping some parameters **fixed** to some predefined values.
- The latter can be done using the function **fixParam()**. Based on the previous fitting, we decide to fix the second shape parameter to one.
- The default (faster) **BFGS** optimization routine also happens to lead to convergence in this case:

```
1 > (fk2 <- fitCopula(khoudrajiCopula(copula2 = gumbelCopula(),
2                               shapes = fixParam(c(NA_real_, 1),
3                                                 c(FALSE, TRUE))),
4                               data = U, start = c(1.1, 0.5)))
```

```
Call: fitCopula(copula, data = data, start = ..1)
Fit based on "maximum pseudo-likelihood" and 659 2-dimensional observations.
Copula: khoudrajiExplicitCopula
c2.alpha    shape1
 2.2729    0.9218
The maximized loglikelihood is 281.9
Optimization converged
```

- Comparing the last two outputs, we see that the difference between the two- and three-parameter models is **very small** in terms of parameter estimates.
- The contour plot of the fitted two-parameter Khoudraji–Gumbel–Hougaard copula **overlaid** by those of the empirical copula for the danube data set is represented in Figure 5.3:

```
1 > cpK <- contourplot2(fk2@copula, FUN = pCopula, region = FALSE,
2                         key = list(corner = c(0.04, 0.04),
3                                 lines = list(col = 1:2, lty = 1),
4                                 text = list(c("Fitted Khoudraji-Gumbel-Hougaard copula",
5                                         "Empirical copula"))))
```



```
6 > cpK + cpCn
```

- Whether the Khoudraji–Gumbel–Hougaard family **fits better** than the Gumbel–Hougaard family is orangeunclear at this point and will be investigated in the next example, as well as later in this chapter based on formal inference procedures.



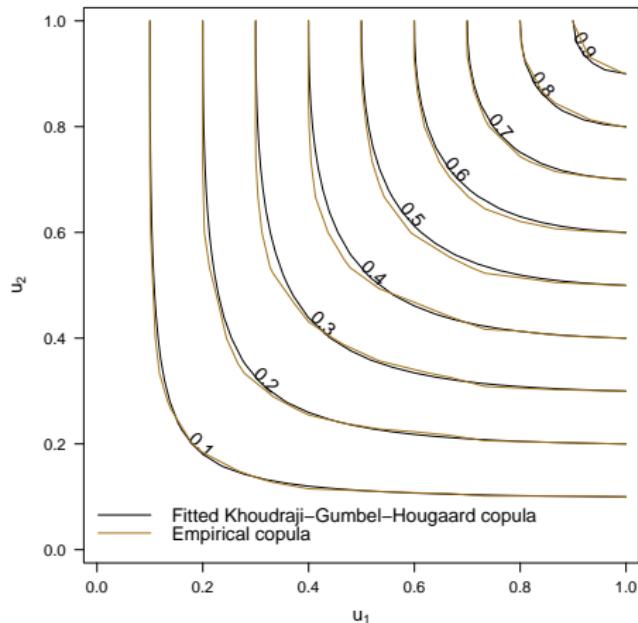


Figure 5.3 Contour plot of the fitted Khoudraji–Gumbel–Hougaard copula overlaid by those of the empirical copula for the danube data set.

5.2 Hypothesis tests

- Although plots and graphical diagnostics are crucial in any data analysis, they should be used with care if they do not incorporate in some way the influence of sampling variability.
- A prominent example is that of a Q-Q plot without confidence bands as there is no objective way to decide between accepting or rejecting a fitted univariate distribution in the case of a moderate departure from the line $y = x$.
- What a “small” or “large” departure from this line is depends in general on the type of problem at hand and in particular on the sample size.
- In the rest of this section, we review selected formal statistical tests, that is, hypothesis tests which compute p-values.
- Such tests can help to guide the choice of the hypothesized copula family \mathcal{C} .

5.2.1 Tests of independence

- In the setting under consideration, a first natural question is whether C is actually **different** from the **independence copula** Π .
- If it is not, only **marginal modeling** is **necessary**, which can be carried out using classical statistical approaches for univariate iid observations.
- Formally, this amounts to testing

$$\mathcal{H}_0 : C = \Pi \quad \text{versus} \quad \mathcal{H}_1 : C \neq \Pi. \quad (5.2)$$

- In dimension two, a classical **alternative** consists of testing

$$\mathcal{H}_0 : \tau = 0 \quad \text{versus} \quad \mathcal{H}_1 : \tau \neq 0,$$

or

$$\mathcal{H}_0 : \rho_s = 0 \quad \text{versus} \quad \mathcal{H}_1 : \rho_s \neq 0.$$

- From Chapter 2, we know that $C = \Pi$ implies $\tau = 0$ and $\rho_s = 0$, but that the converse is false in general.

- This lack of equivalence, however, is usually not an issue in practice as copulas $C \neq \Pi$ such that $\tau = 0$ or $\rho_s = 0$ do not seem to arise often in applications.
- In any case, the latter configuration can be easily discarded by a scatter plot of the observations.

Example 5.2.1 (Test of uncorrelatedness)

We consider again the danube data set presented initially in Example 4.1.6 and test formally for the uncorrelatedness in terms of Kendall's tau between the two underlying variables:

```
1 > data(danube, package = "lcopula")
2 > cor.test(danube[,1], danube[,2], method = "kendall")
```

Kendall's rank correlation tau

```
data: danube[, 1] and danube[, 2]
z = 21.064, p-value < 2.2e-16
alternative hypothesis: true tau is not equal to 0
sample estimates:
tau
```

- As expected from the **graphical diagnostics**, the extremely small p-value provides **strong evidence against** the null hypothesis that $\tau = 0$, and therefore strong evidence in favor of the alternative hypothesis that $\tau \neq 0$, which in turn implies that $C \neq \Pi$.
- The analog test based on Spearman's rho can be carried out by setting `method = "spearman"` in `cor.test()`.

□

Example 5.2.2 (A fallacy of a test of uncorrelatedness)

- Let X be a standard normal random variable and let $Y = X^2$. Clearly, X and Y are **not independent**, so the copula C of (X, Y) is different from the independence copula and yet $\tau = 0$.

- The following code illustrates the **inappropriateness** of the test of uncorrelatedness based on Kendall's tau when carried out on independent observations from (X, Y) :

```
1 > set.seed(1515)
2 > x <- rnorm(200)
3 > y <- x^2
4 > cor.test(x, y, method = "kendall")
```

Kendall's rank correlation tau

```
data: x and y
z = 0.6594, p-value = 0.5096
alternative hypothesis: true tau is not equal to 0
sample estimates:
tau
0.03135678
```

- The large p-value provides **no evidence against** the null hypothesis that $\tau = 0$, which is therefore a plausible hypothesis.
- A scatter plot, however, immediately reveals that $C \neq \Pi$. □

- The tests of uncorrelatedness based on Kendall's tau or Spearman's rho could be adapted for the setting $d > 2$ by considering the higher-dimensional extensions of these quantities studied in Joe (1990) or Schmid and Schmidt (2007).
- A more **direct** way of testing \mathcal{H}_0 in (5.2) when $d \geq 2$ consists of assessing the difference between a **non-parametric estimate** of C and an **estimate** of the latter **under \mathcal{H}_0** .
- From Section 4.2.1, a natural non-parametric estimator of C is the **empirical copula**, while an obvious estimator of C under \mathcal{H}_0 is Π , leading for instance to the **test statistic**

$$S_n^{\Pi} = n \int_{[0,1]^d} (C_n(\mathbf{u}) - \Pi(\mathbf{u}))^2 d\mathbf{u}. \quad (5.3)$$

- Such a test was initially suggested by Deheuvels (1981) and was thoroughly revisited in Genest and Rémillard (2004) and Genest et al. (2007).

- In particular, the asymptotics of S_n^{Π} under \mathcal{H}_0 in (5.2) follow immediately from those of the empirical copula process (4.11).
- From a practical perspective, an approximate p-value for S_n^{Π} can be obtained by simulation.
- Given a large integer N , say $N = 1000$, independent realizations $S_n^{\Pi,(1)}, \dots, S_n^{\Pi,(N)}$ of S_n^{Π} under \mathcal{H}_0 are simulated as explained in Genest and Rémillard (2004), and an approximate p-value is computed as

$$\frac{1}{N+1} \left(\sum_{k=1}^N \mathbf{1}(S_n^{\Pi,(k)} \geq S_n^{\Pi}) + \frac{1}{2} \right).$$

- The above slight modification of the classical formula

$$N^{-1} \sum_{k=1}^N \mathbf{1}(S_n^{\Pi,(k)} \geq S_n^{\Pi})$$

is used to ensure that the p-value is in the open interval $(0, 1)$ so that transformations by quantile functions of continuous distributions are always well-defined.

- This approach is adopted in the R package `copula` for all tests for which approximate p-values are computed using **simulation** or **resampling**, the default value for N being 1000.
- Consequently, for such tests, approximate p-values are numbers in the set $\{(0 + 1/2)/(N + 1), \dots, (N + 1/2)/(N + 1)\}$.

Example 5.2.3 (Test of independence based on S_n^{Π})

Assume that C is C_2^F , the Frank copula with parameter 2, and let us apply the **test of independence** studied in Genest and Rémillard (2004) to a sample of size n from any meta-Frank distribution constructed from C_2^F :

```

1 > n <- 100
2 > d <- 3
3 > set.seed(1969)
4 > U <- rCopula(n, frankCopula(2, dim = d))

```

To carry out the test, we first **generate 1000 realizations** of the test statistic under independence for sample size n and dimension d using the

function `indepTestSim()` and then compute the approximate p-value using `indepTest()`:

```
1 > dist <- indepTestSim(n, p = d, verbose = FALSE)
2 > indepTest(U, d = dist)
```

```
Global Cramer-von Mises statistic: 0.1295138 with p-value 0.0004995005
Combined p-values from the Mobius decomposition:
  0.0004995005  from Fisher's rule,
  0.008491508  from Tippett's rule.
```

- The first p-value is the p-value for the asymptotically equivalent version of S_n^{Π} considered in Genest and Rémillard (2004). The definitions of the statistics corresponding to the remaining two p-values can be found in the latter reference and in Kojadinovic and Holmes (2009, Section 3.5).
- Hence, as expected, the tests provide **strong evidence against** (respectively, in favor of) the null hypothesis that $C = \Pi$ (respectively, the alternative that $C \neq \Pi$).



5.2.2 Tests of exchangeability

- Many popular parametric copula families are **exchangeable** and it is therefore of interest to assess whether such an hypothesis is **plausible** for the data at hand.
- For $d = 2$, a natural test statistic is

$$S_n^{\text{ex}_C} = n \int_{[0,1]^2} (C_n(u_1, u_2) - C_n(u_2, u_1))^2 dC_n(\mathbf{u}), \quad (5.4)$$

where C_n is the **empirical copula**.

- The corresponding test is studied in Genest et al. (2012) as a particular case of the one studied in Rémillard and Scaillet (2009); see also Kojadinovic and Yan (2012, Section 5).
- An extension to arbitrary dimensions $d > 2$ is considered in Harder and Stadtmüller (2017).

Example 5.2.4 (Test of exchangeability based on $S_n^{\text{ex}_C}$)

The test studied in Genest et al. (2012) is implemented in the R function `exchTest()`. We can easily apply it to the `danube` data set:

```
1 > set.seed(1453)
2 > exchTest(as.matrix(danube))
```

```
Test of exchangeability for bivariate copulas with argument
'm' set to 0
```

```
data: as.matrix(danube)
statistic = 0.051969, p-value = 0.001499
```

- Setting the seed prior to the function call is done to guarantee reproducibility since the p-value for $S_n^{\text{ex}_C}$ is computed using resampling, which involves pseudo-random number generation.
- Looking at the p-value, we conclude that there is strong evidence against exchangeability in the `danube` data set.



Remark 5.2.5

- 1) ■ As with any statistical test, it is important to distinguish between the so-called “presence of the effect” and the “practical importance of the effect”; see, for example, Wild and Seber (1999, Chapter 9).
 - The result of the test indicates that it is **extremely likely** that the underlying unknown copula is **non-exchangeable**.
 - However, it does not say anything about the **strength** of the non-exchangeability.
- 2) ■ We have already carried out two tests on the danube data set, and the results of more tests on the same data set will be presented in the remaining sections of this chapter.
 - Consequently, with the **issue of multiple testing** in mind, rejection or non-rejection of a given null hypothesis on the underlying unknown copula C should not be decided by comparing its p-value with, say, the classical $\alpha = 5\%$ significance level.

- In general, for a study involving m tests, a Bonferroni correction could be used.
- It consists of using α/m as significance level for each test.
- Such a correction is, however, known to lead to a too conservative control of the so-called family-wise error rate when m is large or the test statistics are positively dependent.
- In the case of a study on the danube data set, for instance, carrying out each test, say, at the 1% level might be an acceptable compromise.



5.2.3 A test of radial symmetry

- Similarly to exchangeability, we can also test radial symmetry.

- Starting from the definition of radial symmetry of a copula, see Section 2.5, a natural **test statistic** is

$$S_n^{\text{sym}} = n \int_{[0,1]^d} (C_n(\mathbf{u}) - \bar{C}_n(\mathbf{u}))^2 dC_n(\mathbf{u}), \quad (5.5)$$

where C_n is the **empirical copula** of $\mathbf{X}_1, \dots, \mathbf{X}_n$ and \bar{C}_n is the empirical copula of $-\mathbf{X}_1, \dots, -\mathbf{X}_n$.

- Recall that C_n is merely the **empirical df of the sample of pseudo-observations** $\mathbf{U}_{1,n}, \dots, \mathbf{U}_{n,n}$ defined in (4.8) while some thought reveals that \bar{C}_n is the empirical df of the sample $\mathbf{1} - \mathbf{U}_{1,n}, \dots, \mathbf{1} - \mathbf{U}_{n,n}$.
- The test based on S_n^{sym} is studied in the **bivariate** case by Genest and Nešlehová (2014).
- An alternative **multivariate** version of it is considered in Kojadinovic (2017b).

Example 5.2.6 (Test of radial symmetry based on S_n^{sym})

- The test based on S_n^{sym} is implemented in the R function `radSymTest()`.

- It also relies on **resampling** for computing an approximate p-value.
- To illustrate its **computational cost**, we shall record and report the run time of one execution of `exchEVTest()` on the `danube` data set using the following wrapper `withTime()` of R's `system.time()`:

```
1 > withTime <- function(expr, ...)  
2 {  
3     st <- system.time(r <- expr, ...)  
4     list(value = r, sys.time = st)  
5 }
```

- Applying it to the `danube` data set, we obtain:

```
1 > set.seed(1453)  
2 > withTime(radSymTest(as.matrix(danube)))
```

```
Test of radial symmetry based on the empirical copula
```

```
data: as.matrix(danube)  
statistic = 0.22975, p-value = 0.0004995
```

```
User time: 1.8 sec
```

- We conclude from the p-value that there is **strong evidence against radial symmetry** in the danube data set, which is in accordance with the scatter plot of the pseudo-observations displayed on the left-hand side of Figure 5.1.
- For the rdj data set, we obtain:

```
1 > data(rdj)
2 > Xrdj <- as.matrix(rdj[,-1]) # omitting component 'Date'
3 > set.seed(1389)
4 > withTime(radSymTest(Xrdj))
```

```
Test of radial symmetry based on the empirical copula
```

```
data: Xrdj
statistic = 0.051834, p-value = 0.521
```

```
User time: 7.9 sec
```

- There is thus **no evidence against radial symmetry** in the rdj data set.



5.2.4 Tests of extreme-value dependence

- Maybe somewhat surprisingly, **extreme-value copulas** do not only arise in applications when the available data $\mathbf{X}_1, \dots, \mathbf{X}_n$ are **componentwise maxima** as is sometimes the case in hydrology, climatology or meteorology for instance.
- It is thus of practical interest to have tests to assess whether the assumption of **extreme-value dependence** for C in (5.1) is plausible.
- More formally, the underlying null and alternative hypotheses are

$$\mathcal{H}_0 : C \in \mathcal{E} \quad \text{against} \quad \mathcal{H}_1 : C \notin \mathcal{E},$$

where \mathcal{E} denotes the set of **extreme-value copulas**.

- **Three tests** implemented in the R package `copula`: see the functions `evTestK()`, `evTestA()` and `evTestC()`.
- Note that a recent review of tests of extreme-value dependence can be found in Bücher and Kojadinovic (2015).

5.2.5 Goodness-of-fit tests

- The classes of tests described in the preceding sections can be used to **narrow down** the search for an **adequate parametric copula family**.
- For each family \mathcal{C} that was not previously rejected, it is then typically of interest to assess **whether the unknown copula C actually belongs to \mathcal{C}** .
- This **goodness-of-fit issue** amounts formally to testing

$$\mathcal{H}_0 : C \in \mathcal{C} \quad \text{versus} \quad \mathcal{H}_1 : C \notin \mathcal{C}.$$

- A large number of goodness-of-fit tests were proposed in the literature as can be concluded for instance from the recent reviews of Genest et al. (2009a) and Fermanian (2013).
- Among the existing procedures, so-called **blanket tests** require neither an arbitrary categorization of the data, nor any strategic choice of smoothing parameter, weight function, kernel, window, etc.

- Among blanket tests, one approach that appears to perform particularly well according to the large scale simulations carried out in Genest et al. (2009a) and Berg (2009) is based on the [empirical copula](#) C_n .
- The empirical copula is a [consistent estimator](#) of the unknown copula C [whether \$\mathcal{H}_0\$ is true or not](#).
- Hence, as suggested in Fermanian (2005), Quessy (2005), and Genest and Rémillard (2008), a natural [goodness-of-fit test](#) consists of comparing C_n with an estimate C_{θ_n} of C obtained under the assumption that $C \in \mathcal{C}$ holds.
- In the previous statement, θ_n is an estimator of θ computed from the pseudo-observations $U_{1,n}, \dots, U_{n,n}$ defined in (4.8) such as the maximum pseudo-likelihood estimator.
- According to the large scale simulations carried out in Genest et al. (2009a), the [most powerful version](#) of this procedure is based on the

Cramér-von Mises statistic

$$\begin{aligned} S_n^{\text{gof}} &= \int_{[0,1]^d} n(C_n(\mathbf{u}) - C_{\theta_n}(\mathbf{u}))^2 dC_n(\mathbf{u}) \\ &= \sum_{i=1}^n (C_n(\mathbf{U}_{i,n}) - C_{\theta_n}(\mathbf{U}_{i,n}))^2. \end{aligned} \quad (5.6)$$

- An **approximate p-value** for the test based on the above statistic can be obtained by means of a *parametric bootstrap* whose asymptotic validity is investigated in Genest and Rémillard (2008).

Algorithm 5.2.7 (Parametric bootstrap)

- 1) Compute the **pseudo-observations** $\mathbf{U}_{1,n}, \dots, \mathbf{U}_{n,n}$ defined in (4.8).
- 2) Compute an estimate $\boldsymbol{\theta}_n$ of $\boldsymbol{\theta}$ from the pseudo-observations $\mathbf{U}_{1,n}, \dots, \mathbf{U}_{n,n}$.
- 3) Compute the test statistic S_n^{gof} .
- 4) For some large integer N , **repeat** the following steps **for every** $k \in \{1, \dots, N\}$:

- 4.1) Generate a pseudo-random sample $\mathbf{U}_1^{(k)}, \dots, \mathbf{U}_n^{(k)}$ from the fitted copula C_{θ_n} and compute the corresponding pseudo-observations $\mathbf{U}_{1,n}^{(k)}, \dots, \mathbf{U}_{n,n}^{(k)}$.
- 4.2) Compute an estimate $\boldsymbol{\theta}_n^{(k)}$ of $\boldsymbol{\theta}$ from the pseudo-observations $\mathbf{U}_{1,n}^{(k)}, \dots, \mathbf{U}_{n,n}^{(k)}$ using the same (rank-based) estimator as in Step 2).
- 4.3) Compute the corresponding version $S_n^{\text{gof},(k)}$ of S_n^{gof} as:

$$S_n^{\text{gof},(k)} = \sum_{i=1}^n \left(C_n^{(k)}(\mathbf{U}_{i,n}^{(k)}) - C_{\boldsymbol{\theta}_n^{(k)}}(\mathbf{U}_{i,n}^{(k)}) \right)^2,$$

where

$$C_n^{(k)}(\mathbf{u}) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}(\mathbf{U}_{i,n}^{(k)} \leq \mathbf{u}), \quad \mathbf{u} \in [0, 1]^d.$$

Under \mathcal{H}_0 , $S_n^{\text{gof},(k)}$ can be thought of as an approximately independent copy of S_n^{gof} .

- 5) An approximate p-value for the test is given by $(1/2 + \sum_{k=1}^N 1(S_n^{\text{gof},(k)} \geq S_n^{\text{gof}}))/(N + 1)$.

□

- The main advantage of the above procedure is its conceptual simplicity.
- The goodness-of-fit of a parametric family \mathcal{C} can be assessed as long as one knows how to estimate the unknown parameter vector θ and how to generate pseudo-random samples from C_θ .
- If C_θ cannot be evaluated, the procedure can be adapted by adding a “second-level” bootstrap; see Genest and Rémillard (2008).
- Similar parametric bootstrap-based procedures can be implemented for many other goodness-of-fit statistics; for reviews, see, for example, Genest et al. (2009a) and Berg (2009).

Example 5.2.8 (Parametric bootstrap-based tests)

- The key R function for carrying out goodness-of-fit tests in the package `copula` is called `gofCopula()`.
- By default, it is based on the statistic S_n^{gof} and implements the above **parametric bootstrap**.
- We can then assess, for instance, whether the Clayton copula family is a good choice for the `rdj` data set introduced in Example 4.1.3:

```
1 > set.seed(1598)
2 > withTime(gofCopula(claytonCopula(dim = 3), x = Xrdj, optim.method="BFGS"))
```

```
Parametric bootstrap-based goodness-of-fit test of Clayton
copula, dim. d = 3, with 'method'="Sn", 'estim.method'="mpl":
```

```
data: x
statistic = 0.87338, parameter = 0.58565, p-value = 0.0004995
```

```
User time: 76.6 sec
```

- Estimation of the unknown parameter of the Clayton copula is carried out by maximum pseudo-likelihood estimation, which is the default estimation method.
- The underlying numerical optimization routine can be changed via the argument `optim.method` passed to `fitCopula()`, which is set to "BFGS" in the code above to avoid a few warnings related to possible convergence problems in this case.
- As one can see from the output, the parameter estimate is $\theta_n = 0.58565$, the test statistic is $S_n^{\text{gof}} = 0.87338$ and the corresponding p-value (computed as in Algorithm 5.2.7 with $N = 1000$) is 0.0004995 , which corresponds to $0.5/1001$.
- As could have been expected given the fact that the observations under consideration are trivariate financial log-returns, the Clayton family is strongly rejected.



- The **main inconvenience** of the above goodness-of-fit test based on the parametric bootstrap is its **high computational cost**, as each iteration requires both random number generation from the fitted copula and estimation of the copula parameters.
- As the sample size increases, its **application becomes prohibitive**.
- In order to circumvent this high computational cost, a **faster**, large-sample testing procedure based on **multiplier central limit theorems** was proposed in Kojadinovic et al. (2011).
- As n reaches 300, the Monte Carlo experiments carried out for $d = 2, 3$ and 4 in Kojadinovic and Yan (2011a) indicate that one can safely use the multiplier approach as an alternative to the parametric bootstrap.
- The previous statement means that **under \mathcal{H}_0 both resampling procedures are asymptotically equivalent** and thus asymptotically valid.
- Under the alternative, however, they might not necessarily return similar p-values.

- The price to pay for the higher computational efficiency of the multiplier resampling approach is more programming work as certain partial derivatives need to be computed for each hypothesized parametric copula family \mathcal{C} .
- For those families in copula for which these partial derivatives are not implemented, numerical differentiation based on the R package numDeriv of Gilbert and Varadhan (2016) is used.

Example 5.2.9 (Multiplier goodness-of-fit tests)

For the rdj data set, given the fact that the sample size is large, the multiplier resampling approach can be safely used as an alternative to the default parametric bootstrap by utilizing `gofCopula(, simulation = "mult")`:

```
1 > set.seed(1610)
2 > withTime(gofCopula(claytonCopula(dim = 3), x = Xrdj, simulation = "mult"))
```

```
Multiplier bootstrap-based goodness-of-fit test of Clayton
copula, dim. d = 3, with 'method'='Sn', 'estim.method'='mpl':
```

```
data: x
statistic = 0.87338, parameter = 0.58565, p-value = 0.0004995
User time: 4.4 sec
```

As previously, the **Clayton family** is **strongly rejected**. We can also test whether the **normal family** fits:

```
1 > set.seed(1685)
2 > gofCopula(normalCopula(dim = 3, dispstr = "un"), x = Xrdj,
3           simulation = "mult")
```

```
Multiplier bootstrap-based goodness-of-fit test of Normal
copula, dim. d = 3, with 'method'="Sn", 'estim.method'="mpl":
```

```
data: x
statistic = 0.04101, parameter1 = 0.57812, parameter2 =
0.34002, parameter3 = 0.40169, p-value = 0.09041
```

A similar hypothesis for the ***t* family** in which the degrees of freedom are kept fixed at 10 can be tested as follows:

```
1 > set.seed(1792)
2 > gofCopula(tCopula(dim = 3, dispstr = "un", df.fixed = TRUE, df = 10),
3           x = Xrdj, simulation = "mult")
```

```
Multiplier bootstrap-based goodness-of-fit test of t-copula,
dim. d = 3, with 'method'='Sn', 'estim.method'='mpl':
```

```
data: x
statistic = 0.022475, parameter1 = 0.59238, parameter2 =
0.36251, parameter3 = 0.42644, p-value = 0.6538
```

Hence, neither the normal family nor the t family with 10 degrees of freedom are rejected. Both assumptions on \mathcal{C} are thus **plausible**. The fact that the p-value for the t family with 10 degrees of freedom is higher should not in principle be used as an argument to select it. □

Example 5.2.10 (Empirical levels of the multiplier goodness-of-fit test for the Joe family)

- The **multiplier** goodness-of-fit test is expected to be a **valid large-sample alternative** to the **parametric bootstrap**.
- The following code shows how the user can **empirically assess** whether such a test **holds its level** in the case of small samples, for instance, when assessing the fit of the Joe family:

```
1 > theta <- iTau(joeCopula(), tau = 0.5) # Joe copula parameter
2 > ##' @title P-value of multiplier goodness-of-fit test on data
3 > ##' generated under the null hypothesis
4 > ##' @param n sample size
5 > ##' @param theta Joe copula parameter
6 > ##' @return p-value of the multiplier goodness-of-fit test
7 > pvalMult <- function(n, theta)
8 {
9     U <- rCopula(n, copula = joeCopula(theta))
10    gofCopula(joeCopula(), x = pobs(U), simulation = "mult",
11               optim.method = "BFGS")$p.value
12 }
```

- Let us then compute, say, 1000 p-values under \mathcal{H}_0 :

```
1 > set.seed(1940)
2 > pv <- withTime(replicate(1000, pvalMult(n = 100, theta = theta)))
```

- Recall that, because of the use of `withTime()`, the output of `replicate()` is stored in `pv$value`, while the corresponding run time is stored in `pv$sys.time`:

```
1 > pv$sys.time # the run time
```

```
User time: 34.1 sec
```

- If the test holds its level, the **p-values** should be **approximately uniformly distributed**.
- The lower tail is of particular interest in practice:

```
1 > alpha <- c(0.01, 0.05, 0.1) # nominal levels
2 > rbind(nom.level = alpha, emp.level = ecdf(pv$value)(alpha)) # levels
```

```
[,1] [,2] [,3]
nom.level 0.010 0.050 0.100
emp.level 0.008 0.044 0.096
```

- It thus seems that the multiplier goodness-of-fit test based on maximum pseudo-likelihood estimation for the Joe family **holds its level** reasonably well when applied to small samples with moderate dependence.
- Before drawing a final conclusion, other values of Kendall's tau should be considered as well.
- The previous code can actually be **adapted** to assess whether any test of interest available in R holds its level.



When $d = 2$ and \mathcal{C} is a one-parameter copula family, **method-of-moment estimation** based on Kendall's tau or Spearman's rho can be used as an alternative to maximum pseudo-likelihood estimation and often results in **faster** goodness-of-fit tests.

Example 5.2.11 (Goodness-of-fit tests based on method-of-moments estimation)

- Let us consider again the `danube` data set presented initially in Example 4.1.6.
- As can be verified, there is no evidence against extreme-value dependence for these observations, which suggests to consider a copula family of the extreme-value type as a possible model.
- Although, from Example 5.2.4, there is strong evidence against exchangeability, for pedagogical purposes, we shall still assess the fit of the Gumbel–Hougaard family.
- To this end, we first apply the parametric bootstrap based on the statistic S_n^{gof} in (5.6) and then do the same with the multiplier bootstrap.
- Instead of using maximum pseudo-likelihood estimation, we use method-of-moments estimation based on Kendall's tau via `gofCopula(, estim.method = "itau")`:

```
1 > Xd <- as.matrix(danube)
2 > set.seed(1613)
3 > gofCopula(gumbelCopula(), x = Xd, estim.method = "itau")
```

```
Parametric bootstrap-based goodness-of-fit test of Gumbel
copula, dim. d = 2, with 'method'='Sn',
'estim.method'='itau':
```

```
data: x
statistic = 0.024399, parameter = 2.2147, p-value = 0.05145
```

```
1 > set.seed(1914)
2 > gofCopula(gumbelCopula(), x = Xd, estim.method = "itau",
3               simulation = "mult")
```

```
Multiplier bootstrap-based goodness-of-fit test of Gumbel
copula, dim. d = 2, with 'method'='Sn',
'estim.method'='itau':
```

```
data: x
statistic = 0.024399, parameter = 2.2147, p-value = 0.07343
```

- As one can see, both versions of the test based on S_n^{gof} give similar p-values providing only **weak evidence against** the Gumbel–Hougaard

family.

- The Joe family, however, is **strongly rejected**:

```
1 > set.seed(1848)
2 > gofCopula(joeCopula(), x = Xd, estim.method = "itau")
```

```
Parametric bootstrap-based goodness-of-fit test of Joe
copula, dim. d = 2, with 'method'="Sn",
'estim.method'="itau":
```

```
data: x
statistic = 0.064507, parameter = 3.2713, p-value = 0.0004995
```



Example 5.2.12 (Goodness of fit of the Khoudraji–Gumbel–Hougaard family)

- Because of the possible slight non-exchangeability in the danube data set (see Figure 5.1) and given that it is plausible that the underlying copula is of the extreme-value type after adequate testing, it is meaningful to

consider a **non-exchangeable extreme-value copula family** as candidate family for the **danube** data set.

- The **Khoudraji–Gumbel–Hougaard** $kho_s(\Pi, C_\theta^{\text{GH}})$ whose df is given by (3.3) is such a family and was already considered in Example 5.1.2 in its two-parameter version.
- For such a **multiparameter model**, method-of-moments estimation based on Kendall's tau or Spearman's rho cannot be applied, which is why default **maximum pseudo-likelihood** estimation is used:

```
1 > set.seed(1969) ## Parametric bootstrap-based test
2 > withTime(
3     gofCopula(khoudrajiCopula(copula2 = gumbelCopula(),
4                               shapes = fixParam(c(NA_real_, 1), c(FALSE, TRUE))),
5                               start = c(1.1, 0.5), x = Xd, optim.method = "Nelder-Mead")
6 )
```

Parametric bootstrap-based goodness-of-fit test of Khoudraji copula, dim. d = 2, constructed from Independence copula Gumbel copula, with 'method'='Sn', 'estim.method'='mpl':

data: x

```
statistic = 0.029712, parameter1 = 2.2729, parameter2 =
0.9218, p-value = 0.04745
```

```
User time: 86.1 sec
```

```
1 > set.seed(1969) ## Multiplier-based test
2 > withTime(
3     gofCopula(khoudrajiCopula(copula2 = gumbelCopula(),
4                             shapes = fixParam(c(NA_real_, 1), c(FALSE, TRUE))),
5     start = c(1.1, 0.5), x = Xd, optim.method = "Nelder-Mead",
6     simulation = "mult")
7 )
```

```
Multiplier bootstrap-based goodness-of-fit test of Khoudraji
copula, dim. d = 2, constructed from Independence copula
Gumbel copula, with 'method'='Sn', 'estim.method'='mpl':
```

```
data: x
statistic = 0.029712, parameter1 = 2.2729, parameter2 =
0.9218, p-value = 0.04046
```

```
User time: 1.2 sec
```

- Both p-values provide only **weak evidence against** the two-parameter Khoudraji–Gumbel–Hougaard family, which could thus be regarded as a

plausible model along with the simpler Gumbel–Hougaard family.

- Model selection (ranking) will be discussed in the following section.



- Notice finally that alternatives to the aforementioned goodness-of-fit statistics are available in the R package copula.
- Most rely on the parametric bootstrap for the computation of approximate p-values.
- More details can be found in the help files of the functions `gofCopula()` and `gofTstat()`.

5.3 Model selection

- Goodness-of-fit testing is a key step when searching for an acceptable parametric copula family for the data at hand.
- Its results can, however, be disappointing in practice.
- When the sample size is small, it is not uncommon that none of the candidate parametric copula families are rejected, meaning that all candidate models are plausible.
- On the contrary, when the dimension or the sample size is large, it can easily happen that all the candidate families are rejected.
- While a goodness-of-fit test's aim is to assess whether a given parametric copula family fits the data at hand, the role of model selection in the setting considered in this chapter is to rank the candidate copula families, resulting, in practice, in the selection of the best ranked family.

- Ideally, one would not want the selected family to be rejected by a goodness-of-fit test but this can obviously happen in applications.
- In the classical likelihood-based setting of parametric statistics, a famous criteria for model selection is the *Akaike information criterion (AIC)*.
- In the “full” likelihood estimation context of Section 4.1.1 in which the **marginal** and **copula** parameters are **estimated simultaneously**, its use is thus **fully justified**.
- In that case, the formula is

$$\text{AIC} = 2(\ell_{n,\max} - p), \quad (5.7)$$

where $\ell_{n,\max}$ is the maximized likelihood and p is the total number of marginal and copula parameters.

- **Many authors**, such as Chen and Fan (2005), McNeil et al. (2015) or Joe (2014) also **apply** the above **AIC** formula in the **two stage estimation contexts** of the inference functions for margins method described in

Section 4.1.1 or the maximum pseudo-likelihood approach presented in Section 4.1.2.

- In the latter case for instance, $\ell_{n,\max}$ is the maximized pseudo-likelihood and p is the number of copula parameters.
- The first and only [theoretical investigation](#) of [model selection](#) for [copula families](#) seems to be due to Grønneberg and Hjort (2014).
- The authors conclude that the use of the classical [AIC](#) formula is [not justified](#) in the context of the previously mentioned [two-stage estimation methods](#), except under restrictive conditions not satisfied by most parametric copula families of interest.
- They propose several alternative selection criteria which do not suffer from the theoretical deficiencies of AIC in the context under consideration.
- One of these criteria, namely the [cross-validation copula information criterion](#) defined in Grønneberg and Hjort (2014, Equation (44)), is

compared with the naive use of AIC in the context of maximum pseudo-likelihood estimation in a bivariate numerical experiment involving five common one-parameter copula families in Jordanger and Tjøstheim (2014).

- The conclusion of the study is that the two selection criteria do not seem to differ significantly overall.
- Given the computational cost of the cross-validation copula information criterion, Jordanger and Tjøstheim (2014) suggest to **keep using the significantly simpler AIC** formula in the context of maximum pseudo-likelihood estimation.
- One should, however, **remain cautious** because the aforementioned numerical study, despite its quality, considers neither some of the pathological examples put forward in Grønneberg and Hjort (2014), nor multivariate multiparameter copula families.

- As explained in Grønneberg and Hjort (2014, Section 4), the aforementioned cross-validation copula information criterion is, up to a multiplicative constant, the first-order equivalent of the [cross-validation criterion](#)

$$\widehat{\text{cv}}_n = \frac{1}{n} \sum_{i=1}^n \log c_{\theta_{n,-i}}(\mathbf{F}_{n,-i}(\mathbf{X}_i)), \quad (5.8)$$

where $\theta_{n,-i}$ is the maximum pseudo-likelihood estimate computed from the sample $\mathbf{X}_1, \dots, \mathbf{X}_{i-1}, \mathbf{X}_{i+1}, \dots, \mathbf{X}_n$ and

$$\mathbf{F}_{n,-i}(\mathbf{x}) = (F_{n,1,-i}(x_1), \dots, F_{n,d,-i}(x_d)), \quad \mathbf{x} \in \mathbb{R},$$

with

$$F_{n,j,-i}(x) = \begin{cases} \frac{1}{n} \sum_{\substack{k=1 \\ k \neq i}}^n 1(X_{kj} \leq x), & \text{if } x \geq \min_{k \in \{1, \dots, n\} \setminus \{i\}} X_{kj}, \\ 1/n, & \text{otherwise.} \end{cases} \quad (5.9)$$

- The above formula is a [natural adaptation](#) of classical [leave-one-out cross validation](#); see, for example, Claeskens and Hjort (2008, Section 2.9) as

well as Grønneberg and Hjort (2014, Remark 1) for a discussion of (5.9) and possible alternatives.

- By construction, it will **penalize copula families** with **too many parameters** that tend to **overfit**.
- It has the advantage of being **easy to implement** but the inconvenience of being **computationally expensive**.

Example 5.3.1 (Cross-validation for the danube data set)

- The R function implementing (5.8) is **xvCopula()**.
- Note that it actually returns \widehat{nxv}_n to be of the **same order of magnitude as the maximized likelihood**.
- For the **danube** data set, we use it to compare the **Joe** family, the **Gumbel–Hougaard** family, and the two-parameter **Khoudraji–Gumbel–Hougaard** considered in Example 5.2.12:

```

1 > Xdan <- as.matrix(danube)
2 > withTime(xvCopula(joeCopula(), x = Xdan))

[1] 246.112
User time: 43.4 sec

1 > withTime(xvCopula(gumbelCopula(), x = Xdan))

[1] 276.3542
User time: 62.6 sec

1 > withTime(
2   xvCopula(khoudrajiCopula(copula2 = gumbelCopula(),
3                             shapes = fixParam(c(NA_real_, 1),
4                                               c(FALSE, TRUE))),
5   x = Xdan, start = c(1.1, 0.5), optim.method = "Nelder-Mead")
6 )

[1] 278.7986
User time: 68 sec

```

- In terms of leave-one-out cross-validation, the **Khoudraji–Gumbel–Hougaard** family should thus be **slightly preferred** to the Gumbel–Hougaard family, which in turn should be preferred to the Joe family.

- To lower the computational cost, a *k*-fold cross-validation analog can be used.
- It consists of dividing the data into $k < n$ blocks of rows of approximately the same size instead of n as in the leave-one-out approach.
- In practice, before forming the blocks, the rows are randomly shuffled, making *k*-fold cross-validation dependent on the seed.

```
1 > k <- 50
2 > set.seed(7)
3 > withTime(xvCopula(joeCopula(), x = Xdan, k = k))
```

```
[1] 246.3862
User time: 3.2 sec
```

```
1 > set.seed(13)
2 > withTime(xvCopula(gumbelCopula(), x = Xdan, k = k))
```

```
[1] 276.5334
User time: 4.7 sec
```

```
1 > set.seed(14)
2 > withTime(
3     xvCopula(khoudrajiCopula(copula2 = gumbelCopula(),
4                               shapes = fixParam(c(NA_real_, 1),
5                                                 c(FALSE, TRUE))),
6     x = Xdan, k = k, start = c(1.1, 0.5),
7     optim.method = "Nelder-Mead")
8 )
```

```
[1] 278.4246
User time: 5 sec
```

- Hence, in the above setting, leave-one-out cross-validation and its k -fold analog induce the **same order** on the candidate copula families, the latter being, however, substantially faster.



Example 5.3.2 (Cross-validation for the rdj data set)

We also compute $\widehat{nXV_n}$ for the `rdj` data set for the **normal** family and the **t** family with 10 degrees of freedom:

```
1 > withTime(xvCopula(normalCopula(dim = 3, dispstr = "un"), x = Xrdj))
```

```
[1] 370.2897
```

```
User time: 274.7 sec
```

```
1 > withTime(xvCopula(tCopula(dim = 3, dispstr = "un", df = 10, df.fixed = TRUE),  
2 x = Xrdj))
```

```
[1] 411.3878
```

```
User time: 541.3 sec
```

The k -fold cross-validation analog leads to the **same ranking**:

```
1 > set.seed(22)  
2 > withTime(xvCopula(normalCopula(dim = 3, dispstr = "un"), x = Xrdj, k = k))
```

```
[1] 370.1114
```

```
User time: 10.8 sec
```

```
1 > set.seed(4)
2 > withTime(xvCopula(tCopula(dim = 3, dispstr = "un", df = 10, df.fixed = TRUE),
3                     x = Xrdj, k = k))
```

```
[1] 411.5618
User time: 21 sec
```

We can actually even compute these selection criteria for the *t copula* when the **degrees of freedom** are treated as a parameter to be **estimated**. For instance:

```
1 > set.seed(1980)
2 > withTime(xvCopula(tCopula(dim = 3, dispstr = "un"), x = Xrdj, k = k))
```

```
[1] 414.1781
User time: 37.1 sec
```

The previous results suggest to consider the following **estimate of the *t copula*** for the **rdj** data set:

```
1 > summary(fitCopula(tCopula(dim = 3, dispstr = "un"), data = pobs(Xrdj)))
```

```
Call: fitCopula(copula, data = data)
Fit based on "maximum pseudo-likelihood" and 1262 3-dimensional observations.
t-copula, dim. d = 3
    Estimate Std. Error
rho.1    0.5877    0.020
rho.2    0.3593    0.026
rho.3    0.4225    0.025
df       6.5025     NA
The maximized loglikelihood is 419.3
Optimization converged
Number of loglikelihood evaluations:
function gradient
    31          10
```



6 Ties, time series and regression

6.1 Ties

6.2 Selected copula tests and models for time series

6.3 Regression

6.1 Ties

- Let X_1, \dots, X_n be a univariate random sample from a df F . If $X_i = X_j$ for some $i \neq j$, the sample contains *ties*.
- It is well-known that if F is continuous, ties occur with probability zero.
- By extension, we shall say that a d -dimensional data set $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n$ contains *ties* if at least one component sample X_{1j}, \dots, X_{nj} , $j \in \{1, \dots, d\}$, contains *ties*.
- If we assume, as in Chapters 4 and 5, that the available data $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n$ are iid from a df H with *continuous* univariate margins, *ties should therefore not occur*.
- Yet, in practice, because of *rounding* and *lack of measurement precision*, it is *not uncommon* to observe *ties* in a data set arising from the observation of a *truly continuous* random phenomenon.

- Most inference procedures on the underlying unknown copula C described in Chapters 4 and 5 are based on pseudo-observations from C computed from the data using (4.8) which are equivalent to the multivariate scaled ranks (4.9) in the absence of ties.
- Because ranks can be defined in several ways when ties are present, the aforementioned equivalence does not necessarily hold anymore if ties are present.
- The following example briefly presents the most frequently encountered definitions of ranks in the presence of ties.

Example 6.1.1 (Computing ranks in the presence of ties)

- The R function `rank()` can be used to compute ranks from a univariate series.
- The method for computing ranks is specified via the argument `ties.method` which is "average" by default.

- Reassuringly, when there are no ties, all methods lead to the same result.
For instance:

```
1 > set.seed(1979)
2 > (U <- runif(8))

[1] 0.9088996 0.4820053 0.2529599 0.7284594 0.1655928 0.2641590
[7] 0.6108628 0.3877454
```

```
1 > R.avg    <- rank(U) # ties.method = "average"
2 > R.random <- rank(U, ties.method = "random")
3 > R.max    <- rank(U, ties.method = "max")
4 > R.min    <- rank(U, ties.method = "min")
5 > stopifnot(R.random == R.avg, R.max == R.avg, R.min == R.avg)
```

- To simulate lack of measurement precision, we divide the unit interval into *b bins* of equal length and replace each original observation by the center of the bin to which it belongs:

```
1 > b <- 10 # number of bins
2 > (U.ties <- cut(U, breaks = 0:b/b, labels = 0.5:(b - 0.5)/b)) # a factor
```

```
[1] 0.95 0.45 0.25 0.75 0.15 0.25 0.65 0.35
Levels: 0.05 0.15 0.25 0.35 0.45 0.55 0.65 0.75 0.85 0.95
```

- As one can see, the **third** and the **sixth** observations are now both equal to 0.25.
- The default setting **`ties.method = "average"`** then assigns to each of the two observations the **average** of the ranks they would obtain if there were no ties.
- In other words, the value 0.25 being the second largest observation and appearing twice, the ranks assigned to the third and the sixth observations are $(2 + 3)/2 = 2.5$:

```
1 > rank(U.ties) # ties.method = "average"
```

```
[1] 8.0 5.0 2.5 7.0 1.0 2.5 6.0 4.0
```

- Computing **average ranks** (also known as ***mid-ranks***) is the default approach in non-parametric statistics to deal with ties.
- As seen above, two other solutions for assigning ranks to tied observations consist of computing **maximum** or **minimum** ranks instead of average ranks:

```
1 > rank(U.ties, ties.method = "max") # maximum rank for each tied observation
```

```
[1] 8 5 3 7 1 3 6 4
```

```
1 > rank(U.ties, ties.method = "min") # minimum rank for each tied observation
```

```
[1] 8 5 2 7 1 2 6 4
```

- Notice the difference between the two previous outputs in the third and sixth components.
- Yet another solution consists of assigning the ranks of tied observations **pseudo-randomly**, which thus depends on the seed of the underlying pseudo-random number generator:

```
1 > set.seed(8)
2 > rank(U.ties, ties.method = "random")
```

```
[1] 8 5 3 7 1 2 6 4
```

```
1 > set.seed(46)
2 > rank(U.ties, ties.method = "random")
```

```
[1] 8 5 2 7 1 3 6 4
```

- As one can see, the ranks of the two tied observations (the third and the sixth), which, if there were no ties, should have been either 2 and 3 or 3 and 2, are assigned pseudo-randomly.

□

- When applying the same method to the component series of the multivariate random sample X_1, \dots, X_n , the four rank-computing method discussed in the previous example can lead to four different sets of multivariate scaled ranks of the form (4.9).
- Notice that only the use of maximum ranks preserves the equivalence between (4.8) and (4.9).
- Furthermore, the minimum, average or maximum rank method can be regarded as being of the same nature in the sense that, unlike the

random method, all three will result in ties in the multivariate scaled ranks.

Example 6.1.2 (Effect of ties on multivariate scaled ranks)

- To illustrate the effect of ties on multivariate scaled ranks, we first generate independent realizations from a bivariate Clayton copula with parameter $\theta = 5$ and then produce a discretized version of the generated data by proceeding as in Example 6.1.1 for each component sample:

```
1 > n <- 200
2 > set.seed(4809)
3 > U <- rCopula(n, claytonCopula(5))
4 > b <- 10 # number of bins
5 > U.ties <- (apply(U, 2, cut, breaks = 0:b/b, labels = FALSE) - 0.5) / b
```

- Multivariate scaled ranks can then be computed using the function `pobs()` whose argument `ties.method` is passed on to the underlying call of `rank()` and thus controls the rank computation method.
- As illustrated by the following code, the function `pobs()` is indeed

merely a **convenient wrapper** of `rank()`:

```
1 > ties.method <- "max" # can be changed to "min", "average" or "random"
2 > stopifnot(all.equal(pobs(U.ties, ties.method = ties.method),
3                         apply(U.ties, 2, rank, ties.method = ties.method) /
4                         (nrow(U.ties) + 1))) # check
```

- Figure 6.1, produced with the following code, compares the multivariate scaled ranks computed from the **original continuous sample**, the multivariate scaled ranks computed from the **discretized sample** using the default `ties.method = "average"` and the multivariate scaled ranks computed from the **discretized sample** using `ties.method = "random"`:

```
1 > plot(pobs(U), xlab = "", ylab = "")
2 > plot(pobs(U.ties), xlim = 0:1, ylim = 0:1, xlab = "", ylab = "")
3 > set.seed(732)
4 > plot(pobs(U.ties, ties.method = "random"), xlab = "", ylab = "")
```

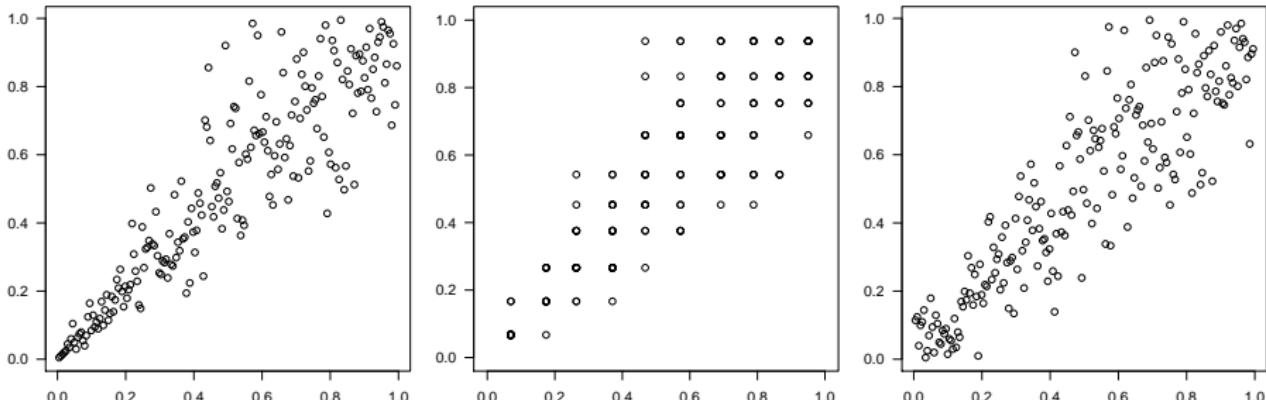


Figure 6.1 Multivariate scaled ranks computed from the original continuous sample (left), from the discretized sample using the default `ties.method = "average"` (middle) and from the discretized sample using `ties.method = "random"` (right).

- As one can see, the **discretization step** (here simulating lack of measurement precision, for instance) results in a **loss of information**.
- While, at first glance, the **right plot** seems to retain **some of the characteristic features** expected from the Clayton copula, it is not difficult to convince oneself that the middle plot could equally well result from the

use of other copulas with a similar Kendall's tau as C_5^C .

- One can also verify that the use of `ties.method = "min"` or `ties.method = "max"` leads to plots of the same nature as the middle one.
- In the context of the current example, note that computing multivariate scaled ranks using `ties.method = "random"` is equivalent to the following:
 - ▶ First, add to each discretized bivariate observation a realization of a bivariate random vector whose components are independent and whose margins are (for instance) $U(-a, a)$ with $0 < a \leq 1/(2b)$,
 - ▶ and then compute bivariate scaled ranks from the resulting sample (which, by construction, does not contain ties anymore).

- This equivalence is illustrated by the following code:

```
1 > a <- runif(1, min = 0, max = 1/(2 * b)) # or a fixed number in (0, 1/(2b))
2 > set.seed(732)
3 > V <- pobs(U.ties + runif(2 * n, min = -a, max = a))
4 > set.seed(732)
5 > stopifnot(all.equal(V, pobs(U.ties, ties.method = "random"))) # check
```

- As a consequence of the previous equivalence, the use of `ties.method = "random"` when computing multivariate scaled ranks does not actually do a better job than the use of `ties.method = "average"`.
- Roughly speaking, absence of information on local dependence due to discreteness is replaced by local independence.
- The latter is best seen by comparing the lower-left corners of the left-hand side and right-hand side plots of Figure 6.1: The lower tail dependence characteristic of Clayton copulas present in the left panel disappeared completely in the right panel.



- The previous example illustrates well the **effect of ties on dependence**: The presence of ties can **hide crucial features** distinguishing one copula family from another.
- The **effect increases with the amount of ties**. As a consequence, inference on the unknown copula C is expected to become more difficult as the proportion of tied observations increases.
- With inference under a parametric assumption on C as in Section 4.1 in mind, a natural question is **how copula estimation is affected by the presence of ties**.
- **Simulations** show that, overall, copula parameter estimation in the presence of ties suffers from a **larger bias** than in the absence of ties.
- The **variability** of the estimation appears to remain **roughly unaffected**, though.

- The effect of the bias can become catastrophic in the case of strong dependence.
- As we have seen in Chapter 5, parametric estimation is often accompanied by graphical diagnostics, formal tests and model selection.
- When dealing with ties, data analysis is even more complicated due to the fact that the vast majority of inference procedures on C proposed in the literature are theoretically studied and implemented under the assumption of no ties in any of the d component series of the given data $\mathbf{X}_1, \dots, \mathbf{X}_n$, that is, no ties in the component samples of the multivariate scaled ranks (4.9).
- A noticeable exception is the test of extreme-value dependence proposed by Cormier et al. (2014, Section 6) which can be adapted for discontinuous margins.
- As far as the inference procedures described in Chapter 5 are concerned, unfortunately, only a few provide meaningful results when applied to

data sets containing ties (unless the proportion of ties is negligible).

- To carry out inference using these tests, there are thus two possible approaches:
 - ▶ (i) transform the original data set so that the resulting multivariate scaled ranks are free of ties, or
 - ▶ (ii) adapt the inference procedures so that they give meaningful results in the presence of ties.
- Approach (i) could be achieved by deleting multivariate observations with at least one tied coordinate before carrying out the inference.
- This rather extreme approach is not advisable as discussed in Genest et al. (2011b, Section 2) since it will not only result in a reduced sample size and thus in less efficient statistical procedures, but it may also alter the underlying dependence structure.

- A second way of carrying out Approach (i) consists of **breaking ties at random**, which is sometimes referred to as *jittering*, that is, adding independent realizations of a small (so that the ranks of untied observations remain unaffected) continuous random variable to all univariate observations.
- As we have already seen in Example 6.1.2, **computing multivariate scaled ranks from jittered data** is equivalent to **computing multivariate scaled ranks** from the original data **using `ties.method = "random"`**.
- Because jittering depends on the seed, the analysis should be performed **multiple times**.
- In the presence of a **moderate amount of ties**, inference on C based on multivariate scaled ranks computed using **`ties.method = "random"`** is considered in Kojadinovic and Yan (2010b, Section 5.1) and Bücher and Kojadinovic (2015, Section 5.2).

- The proposed approach consists of carrying out the inference for **many different seeds** with the hope that the resulting sets of multivariate scaled ranks will not be too different from the pseudo-observations that would have been obtained if there were no ties in the data.
- For a given inference procedure, instead of looking at the average of the results over all seeds, **the distribution of the results is considered**.
- The **motivation** for this stems from the following **heuristic**: If the p-value of a given test is very small (respectively, large) for all seeds, then it is meaningful to reject (respectively, not reject) the corresponding null hypothesis for the original data set.
- For data sets containing a **large proportion of ties**, such as the one generated in Example 6.1.2, the previous approach is **not expected to work** well given the huge number of possible sets of multivariate scaled ranks.

- For a given test, the **set of p-values** obtained from different seeds is thus likely to be **large** and to include both very small and large p-values, thereby making it impossible to draw a conclusion for the original data set according to the aforementioned heuristic.
- An **extension** of the aforementioned method is proposed in [Pappadà et al. \(2017\)](#).
- Let us now turn our attention to **Approach (ii)**. Developing tie-adapted versions of the tests presented in Chapter 5 is difficult.
- This is mostly a consequence of the **difficulty of extending certain key theoretical results** (such as the weak convergence of an appropriate version of the empirical copula process, see Section 4.2.1) from the case of **continuous** margins to the case of **discontinuous** margins.
- Recent **promising results** in that direction are due to Genest et al. (2017) and might allow for a **sound theoretical derivation** of tie-adapted copula inference procedures in the near future.

- A more empirical approach is considered in Kojadinovic (2017b).
- It consists of trying to modify the resampling schemes of existing tests so that they are not too liberal anymore in the presence of ties.
- Such modifications are specifically proposed for the test of exchangeability based on (5.4), the test of radial symmetry based on (5.5), a test of extreme-value dependence, and the parametric bootstrap-based goodness-of-fit test relying on (5.6).
- By construction, the resulting tie-adapted tests should never be too liberal.
- They may, however, be too conservative (and thus lack power) when the sample size is small or the amount of ties is large.
- For the parametric bootstrap-based goodness-of-fit test relying on (5.6), for example, the key idea, initially appearing in Bücher and Kojadinovic (2015), consists of imposing on each parametric bootstrap sample the same “tie structure” as the one found in the original sample.

- Reassuringly, when there are no ties, the modified test leads exactly to the **same computations** as those in Algorithm 5.2.7; see Kojadinovic (2017b, Section 6) for more details.
- Because it relies essentially on **parametric estimation**, one of the few inference procedures on C whose results remain meaningful in the presence of ties is **cross-validation** as implemented in the function `xvCopula()`; see Section 5.3.
- However, one needs to be **careful**. As we have seen earlier, rounding or lack of measurement precision can **hide crucial dependence features**, in particular, in the tails.
- In the presence of ties, **expert knowledge** thus needs to be given even **more weight** than in the absence of ties.
- For instance, if the modeled random phenomenon is known to involve a copula with a certain type of tail dependence, one should **not rely on**

results of the inference if they do not point in the same direction.

- We end this section with the analysis of a well-known data set.

Example 6.1.3 (Analysis of the loss insurance data)

- The Loss/ALAE insurance data are frequently used for illustration purposes in the copula literature; see, for instance, Frees and Valdez (1998), Ben Ghorbal et al. (2009), Kojadinovic and Yan (2010b) and Bücher and Kojadinovic (2015).
- Available in the copula package, the data set consists of 1500 claims of an insurance company.
- Two variables are observed for each claim: loss, an indemnity payment, and alae, the corresponding allocated loss adjustment expense.
- Among the 1500 claims, 34 are censored. As is well-known from survival analysis, ignoring censored observations is likely to lead to biased conclusions.

- Yet, because the inference procedures investigated in this section cannot account for censoring, we shall, **for illustration purposes only**, restrict ourselves to the 1466 uncensored claims here:

```
1 > data(loss)
2 > X <- as.matrix(subset(loss, censored == 0, select = c("loss", "alae")))
```

- Ties affect mostly the **first variable**:

```
1 > ## Percentages of ties
2 > 100 * apply(X, 2, function(x) 1 - length(unique(x))/length(x))
```

	loss	alae
63.096862	4.433834	

- This can also be seen from the corresponding multivariate scaled average ranks and normal scores shown in Figure 6.2:

```
1 > U <- pobs(X)
2 > plot(U, xlab = quote(U[1]~~"(Loss)'), ylab = quote(U[2]~~"(ALAE)"))
3 > Y <- qnorm(U)
4 > plot(Y, xlab = quote(Y[1]~~"(Loss)'), ylab = quote(Y[2]~~"(ALAE)"))
```

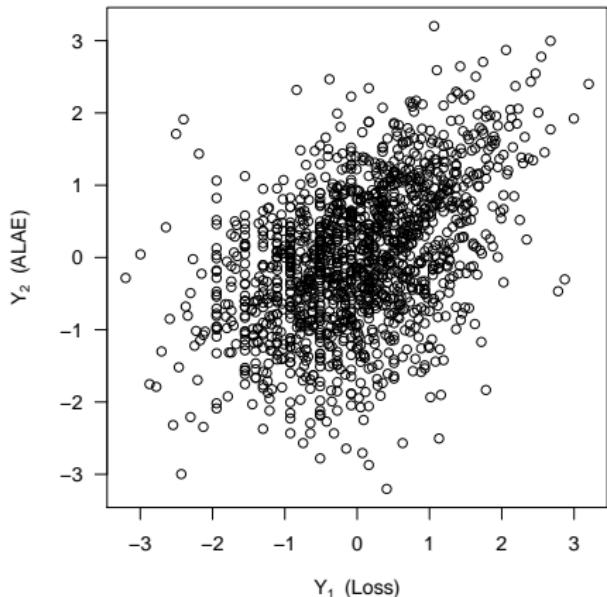
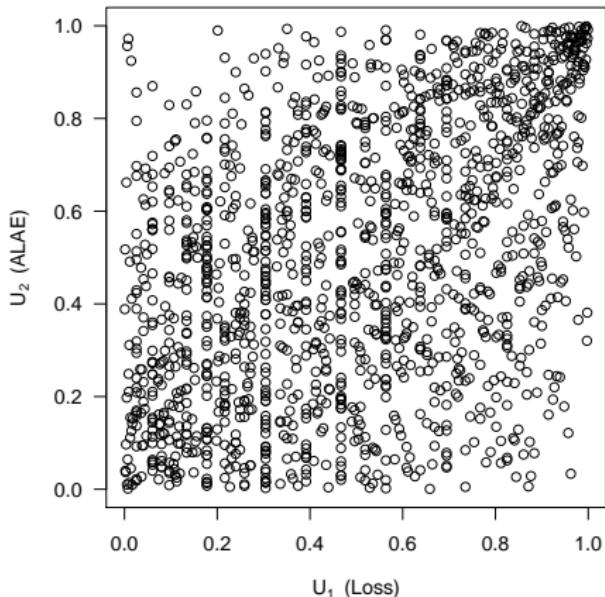


Figure 6.2 Scatter plot of the multivariate scaled average ranks (left) and of the corresponding normal scores (right) of the Loss/ALAE data.

- We start by applying tie-adapted tests of exchangeability, radial symmetry and extreme-value dependence:

```
1 > set.seed(3070)
2 > withTime(round(c(exchTest(X, ties = TRUE)$p.value,
3                     radSymTest(X, ties = TRUE)$p.value,
4                     evTestK(X, ties = TRUE)$p.value), 4))
```

```
[1] 0.0534 0.0005 0.8375
User time: 206.5 sec
```

- There is thus no strong evidence against exchangeability and extreme-value dependence in the data, while there is strong evidence against radial symmetry.
- The above conclusions suggest to consider an exchangeable extreme-value copula as candidate model.
- Given the strong similarity among the existing bivariate parametric exchangeable extreme-value copula families, the Gumbel–Hougaard copula family appears to be a natural choice.

- For illustration purposes, besides the Gumbel–Hougaard family, we also carry out goodness-of-fit testing for the survival Clayton, Frank, Plackett and normal families:

```
1 > ## Goodness-of-fit testing
2 > set.seed(4634)
3 > optim.method <- "BFGS" # the numerical optimization method for MPLE
4 > withTime(
5     round(c(gofCopula(gumbelCopula(), x = X, ties = TRUE,
6                 optim.method = optim.method)$p.value,
7                 gofCopula(rotCopula(claytonCopula()), x = X, ties = TRUE,
8                 optim.method = optim.method)$p.value,
9                 gofCopula(frankCopula(), x = X, ties = TRUE,
10                optim.method = optim.method)$p.value,
11                gofCopula(plackettCopula(), x = X, ties = TRUE,
12                optim.method = optim.method)$p.value,
13                gofCopula(normalCopula(), x = X, ties = TRUE,
14                optim.method = optim.method)$p.value), 4)
15 )
```

```
[1] 0.1683 0.0005 0.0005 0.0005 0.0005
User time: 790.1 sec
```

- Finally, we perform **model selection** based on `xvCopula()`:

```
1 > ## Model selection
2 > set.seed(4807)
3 > k <- 50 # for k-fold cross-validation
4 > withTime(
5     round(c(xvCopula(gumbelCopula(), x = X, k = k,
6             optim.method = optim.method),
7             xvCopula(rotCopula(claytonCopula()), x = X, k = k,
8                     optim.method = optim.method),
9             xvCopula(frankCopula(), x = X, k = k,
10                    optim.method = optim.method),
11             xvCopula(plackettCopula(), x = X, k = k,
12                     optim.method = optim.method),
13             xvCopula(normalCopula(), x = X, k = k,
14                     optim.method = optim.method)), 1)
15 )
```

```
[1] 189.9 182.2 160.4 161.6 169.4
```

```
User time: 27.9 sec
```

- As one can see, the **Gumbel–Hougaard family** is the only one not rejected at the, say, 1% level.
- This extreme-value family is also **ranked first** by the **cross-validation** criterion.

- All of our inference procedures thus suggest to select the Gumbel–Hougaard family as a model for the dependence which is in accordance with the results obtained, for instance, in Chen and Fan (2005), Genest et al. (2006) and Kojadinovic and Yan (2010b).
- Notice that, with `ties = FALSE`, the **p-value** returned by the test of exchangeability based on (5.4) is the **smallest possible** (that is, 0.5/1001) which would strongly suggest to discard the Gumbel–Hougaard family:

```
1 > withTime(exchTest(X, ties = FALSE))
```

```
Test of exchangeability for bivariate copulas with argument  
'm' set to 0
```

```
data: X  
statistic = 0.14727, p-value = 0.0004995
```

```
User time: 2.4 sec
```

- The latter result illustrates again the fact that **presence of ties should not be ignored** when carrying out copula inference. □

6.2 Selected copula tests and models for time series

- Assume that one is given a d -dimensional sample $\mathbf{X}_1, \dots, \mathbf{X}_n$ and considers applying some of the statistical procedures described in Chapters 4 and 5.
- As mentioned therein, for these statistical techniques to provide fully valid results, $\mathbf{X}_1, \dots, \mathbf{X}_n$ need to be iid and their common unknown df H needs to be continuous.
- The iid assumption can fail in various ways. If $\mathbf{X}_1, \dots, \mathbf{X}_n$ cannot be considered as a stretch from a stationary multivariate time series, the use of the inference procedures presented in Chapter 5 on $\mathbf{X}_1, \dots, \mathbf{X}_n$ is meaningless.
- As we shall see, the assumption of stationarity can be formally tested by proceeding, for instance, as in Section 6.2.1.

- Clearly, even if $\mathbf{X}_1, \dots, \mathbf{X}_n$ can be assumed to be a stretch from a stationary multivariate time series, $\mathbf{X}_1, \dots, \mathbf{X}_n$ can still fail to be **serially independent**.
- The presence of **mild serial dependence will not affect**, asymptotically, the **point-estimation** of parameter values when fitting copula families to $\mathbf{X}_1, \dots, \mathbf{X}_n$.
- However, it will result in the **underestimation** of the corresponding **standard errors** as well as in **underestimated approximate p-values** when applying the statistical tests of Chapter 5, which tend to be **too liberal**.
- It is thus **crucial** to be able **to assess** whether $\mathbf{X}_1, \dots, \mathbf{X}_n$ are **serially independent**. approaches to do so will be introduced in Section 6.2.2.
- The fact that the statistical procedures of Chapter 5 should not be applied to $\mathbf{X}_1, \dots, \mathbf{X}_n$ in the presence of serial dependence **may** actually **not be an issue of strong practical importance**.
- When dealing with time series, the interest often lies in **forecasting**.

- To this end, one does not need to model the distribution of the random vector \mathbf{X} of which $\mathbf{X}_1, \dots, \mathbf{X}_n$ are serially dependent copies, but rather the **conditional distribution** of \mathbf{X}_i given the information generated by the past values $\mathbf{X}_1, \dots, \mathbf{X}_{i-1}$.
- A brief overview of the main copula-based models for this conditional distribution will be given in Section 6.2.3 with particular emphasis on **applications in finance and econometrics**.

6.2.1 Tests of stationarity

- The first step prior to any modeling is to assess whether $\mathbf{X}_1, \dots, \mathbf{X}_n$ can be considered as a stretch from a **stationary multivariate time series** $(\mathbf{X}_i)_{i \in \mathbb{Z}}$.
- A time series $(\mathbf{X}_i)_{i \in \mathbb{Z}}$ is said to be **stationary** if, for any $k \in \mathbb{N}$ and $m \in \mathbb{Z}$, the random vector $(\mathbf{X}_1, \dots, \mathbf{X}_k)$ and the random vector $(\mathbf{X}_{1+m}, \dots, \mathbf{X}_{k+m})$ have the same distribution (such a property is

sometimes referred to as strong stationarity in the literature).

- Non-stationarity such as the presence of a trend or a change in the variance could be detected to some extent by plotting the d component series.
- To detect or confirm more subtle departures from stationarity, the use of formal tests is recommended.
- Several genuine tests of stationarity for univariate time series are available in R: The tests of Priestley and Subba Rao (1969), Nason (2013), and Cardinali and Nason (2013) are implemented in the R packages `fractal` of Constantine and Percival (2016), `locits` of Nason (2016) and `costat` of Nason and Cardinali (2013).
- The finite-sample behavior of the first test is particularly disappointing.
- Overall, from the extensive simulations carried out in Bücher et al. (2017), none of the three tests appear to maintain their levels in general, especially when applied to non-Gaussian time series.

- As an imperfect alternative to testing in full generality whether $\mathbf{X}_1, \dots, \mathbf{X}_n$ is a stretch from a stationary time series, one could consider tests particularly sensitive to departures from the hypothesis

$$\mathcal{H}_0 : \text{There exists a df } H \text{ such that } \mathbf{X}_1, \dots, \mathbf{X}_n \text{ have df } H. \quad (6.1)$$

- Such statistical procedures are commonly referred to as *tests for change-point detection*; see, for instance, Csörgö and Horváth (1997) and Aue and Horváth (2013) for an overview of the literature.
- We shall now present a **non-parametric** test with good sensitivity to many alternatives to \mathcal{H}_0 .
- For any $1 \leq k \leq l \leq n$, consider the sub-sample $\mathbf{X}_k, \dots, \mathbf{X}_l$ obtained from $\mathbf{X}_1, \dots, \mathbf{X}_n$ and let

$$H_{k:l}(\mathbf{x}) = \frac{1}{l - k + 1} \sum_{i=k}^l 1(\mathbf{X}_i \leq \mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d,$$

be the empirical df computed from $\mathbf{X}_k, \dots, \mathbf{X}_l$ with the convention that $H_{k:l} = 0$ if $l < k$.

- Following Csörgö and Horváth (1997) and Gombay and Horváth (1999), a **broad class** of non-parametric tests of stationarity particularly sensitive to departures from \mathcal{H}_0 can be derived from the empirical process

$$\mathbb{D}_n^H(t, \mathbf{x}) = \sqrt{n} \lambda_n(0, t) \lambda_n(t, 1) (H_{1:\lfloor nt \rfloor}(\mathbf{x}) - H_{(\lfloor nt \rfloor + 1):n}(\mathbf{x})), \\ (t, \mathbf{x}) \in [0, 1] \times \mathbb{R}^d, \quad (6.2)$$

where $\lambda_n(t, t') = (\lfloor nt' \rfloor - \lfloor nt \rfloor)/n$ for $0 \leq t \leq t' \leq 1$ and where $\lfloor \cdot \rfloor$ denotes the floor function.

- The **normalizing term** $\lambda_n(0, t) \lambda_n(t, 1)$ is typical of cumulative sum **change-point procedures** to which the above class of tests belongs.
- A meaningful **test statistic** is then

$$S_n^H = \sup_{t \in [0, 1]} \int_{\mathbb{R}^d} (\mathbb{D}_n^H(t, \mathbf{x}))^2 dH_{1:n}(\mathbf{x}) = \max_{1 \leq k \leq n-1} S_{n,k}^H, \quad (6.3)$$

where

$$S_{n,k}^H = \frac{1}{n} \sum_{i=1}^n (\mathbb{D}_n^H(k/n, \mathbf{X}_i))^2, \quad (6.4)$$

and was, for instance, used in Holmes et al. (2013) for independent $\mathbf{X}_1, \dots, \mathbf{X}_n$.

- Under stationarity, the difference between $H_{1:k}$ and $H_{k+1:n}$ should be small for all $k \in \{1, \dots, n-1\}$, which implies that S_n^H should be small.
- Large values of S_n^H provide evidence against stationarity.
- The computation of an approximate p-value for the test based on S_n^H relies on a resampling procedure taking into account the strength of the serial dependence in $\mathbf{X}_1, \dots, \mathbf{X}_n$.
- Its asymptotic validity under stationarity and when H in (6.1) has continuous margins can be shown using results from Bücher and Kojadinovic (2016) for short range serial dependence; see also Bücher et al. (2017, Section 2.2).

Example 6.2.1 (Test of stationarity based on S_n^H)

- We begin by testing the **stationarity** of the trivariate time series of **financial log-returns rdj** considered initially in Example 4.1.3.
- To plot the component series against time, we first build an **xts** object representing the trivariate time series with the R package **xts** of Ryan and Ulrich (2017):

```
1 > data(rdj)
2 > library(xts)
3 > Xrdj <- xts(rdj[,-1], order.by = rdj[,1])
```

- Plots of the three component series against time are shown in Figure 6.3 based on **plot.zoo()**:

```
1 > plot.zoo(Xrdj, main = "", xlab = "", mar = c(0, 7, 0, 2.1))
```

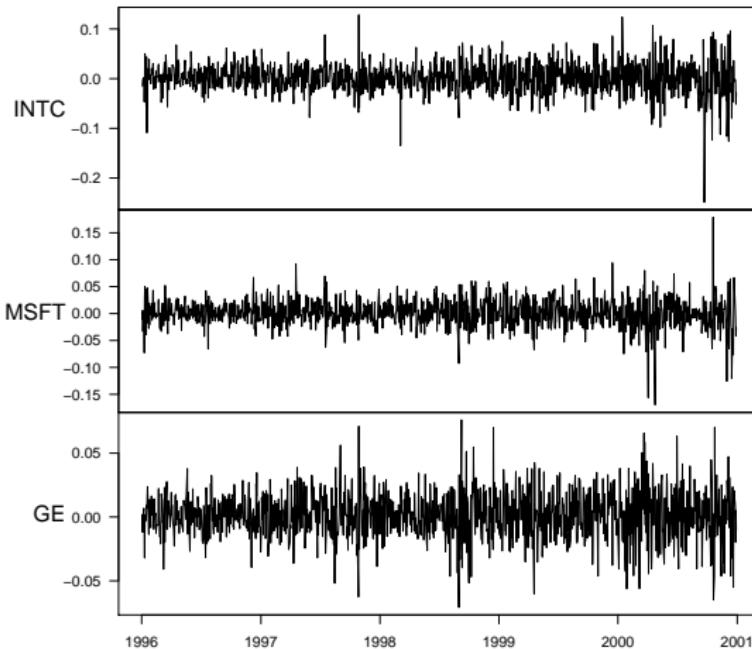


Figure 6.3 Daily log-returns for the Intel (INTC), Microsoft (MSFT) and General Electric (GE) stocks for the period 1996–2000.

- The graphs seem to reveal an **increased volatility** towards the end of the observation period.

- Whether stationarity should be rejected based on these plots is, however, unclear.
- The test based on S_n^H is implemented in the function `cpDist()` of the R package `npcp` of Kojadinovic (2017a).
- The computation of an approximate p-value for S_n^H requires the knowledge of a `bandwidth parameter` corresponding to the argument `b` in `cpDist()`.
- To have `b` `estimated` from the available (possibly serially dependent) data, we set `b` to `NULL`.
- As the test is based on `resampling`, the seed is set to ensure reproducibility:

```
1 > library(npcp)
2 > set.seed(981)
3 > withTime(cpDist(Xrdj, b = NULL))
```

```
Test for change-point detection sensitive to changes in the  
distribution function with 'method'="nonseq"
```

```
data: Xrdj  
cvmmax = 325.12, p-value = 0.002498
```

```
User time: 43.8 sec
```

- The small p-value provides **evidence against stationarity**.
- As a second illustration, we consider a data set initially analyzed in Grégoire et al. (2008).
- It consists of **bivariate daily log-returns** computed from three years of **daily prices** (for the period July 2003 to July 2006) of **crude oil** and **natural gas**:

```
1 > data(gasoil) # oil and gas prices  
2 > library(qrmtools) # for returns()  
3 > Rgasoil <- returns(gasoil[,-1]) # bivariate daily log-returns  
4 > Xgasoil <- xts(Rgasoil, order.by = gasoil[-1, 1]) # corresponding xts object
```

Figure 6.4 shows plots of the two component series against time:

```
1 > plot.zoo(Xgasoil, main = "", xlab = "", mar = c(0, 7, 0, 2.1))
```

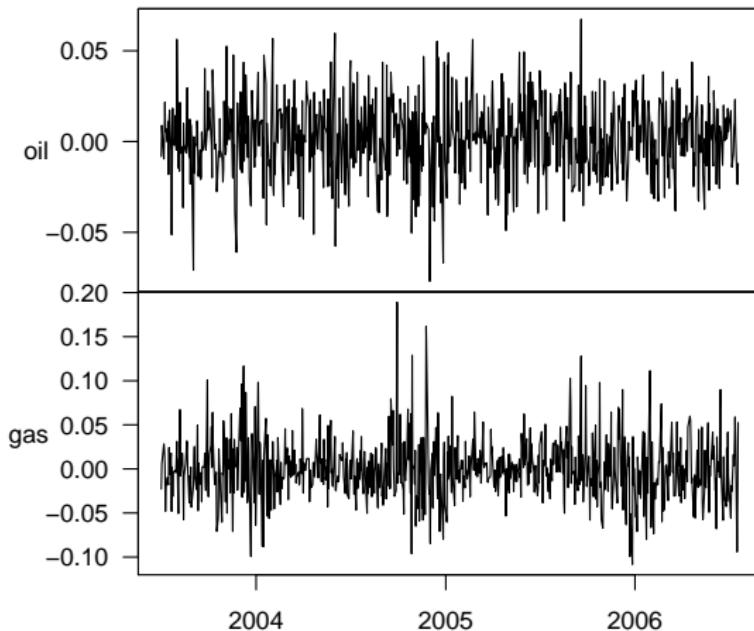


Figure 6.4 Daily log-returns computed from three years of daily prices (from July 2003 to July 2006) of crude oil and natural gas.

- We can now apply the test based on S_n^H as done previously:

```
1 > set.seed(292)
2 > withTime(cpDist(Xgasoil, b = NULL))
```

```
Test for change-point detection sensitive to changes in the
distribution function with 'method'="nonseq"
```

```
data: Xgasoil
cvmmax = 60.212, p-value = 0.3362
```

```
User time: 4.6 sec
```

- The large p-value provides no evidence against stationarity.



- The test based on S_n^H is a non-parametric, general-purpose test which seems to be powerful against alternatives involving changes in the marginal dfs of the observations.
- However, it turns out to be not very sensitive to changes which only affect the underlying copula; see Holmes et al. (2013).

- Assume that \mathcal{H}_0 holds and that, additionally, the common marginal dfs F_1, \dots, F_d of $\mathbf{X}_1, \dots, \mathbf{X}_n$ are continuous.
- By Sklar's Theorem, the common multivariate df H of the observations can be written as

$$H(\mathbf{x}) = C(F_1(x_1), \dots, F_d(x_d)), \quad \mathbf{x} \in \mathbb{R}^d,$$

where C is the unique copula associated with H .

- It follows that \mathcal{H}_0 can be decomposed as $\mathcal{H}_{0,m} \wedge \mathcal{H}_{0,c}$, where

$\mathcal{H}_{0,m}$: There exist F_1, \dots, F_d such that $\mathbf{X}_1, \dots, \mathbf{X}_n$

have marginal dfs F_1, \dots, F_d , (6.5)

$\mathcal{H}_{0,c}$: There exists a copula C such that $\mathbf{X}_1, \dots, \mathbf{X}_n$ have copula C .

(6.6)

- Roughly speaking, the multivariate df changes if and only if the marginal dfs change or the copula changes.

- Let $\neg\mathcal{H}_{0,c}$ denote the negation of $\mathcal{H}_{0,c}$. Several non-parametric tests of stationarity designed to be **particularly sensitive to certain alternatives under $\mathcal{H}_{0,m} \wedge (\neg\mathcal{H}_{0,c})$** are proposed in the literature.
- For instance, tests particularly sensitive to changes in **Kendall's tau** are investigated by Gombay and Horváth (1999), Quessy et al. (2013) and Dehling et al. (2014).
- To derive tests particularly sensitive to changes in the copula, Bücher et al. (2014) consider the following **variant** of the process \mathbb{D}_n^H in (6.2):

$$\mathbb{D}_n^C(t, \mathbf{u}) = \sqrt{n} \lambda_n(0, t) \lambda_n(t, 1) (C_{1:\lfloor nt \rfloor}(\mathbf{u}) - C_{(\lfloor nt \rfloor + 1):n}(\mathbf{u})), \\ (t, \mathbf{u}) \in [0, 1]^{d+1}, \quad (6.7)$$

where, for any $1 \leq k \leq l \leq n$,

$$C_{k:l}(\mathbf{u}) = \frac{1}{l - k + 1} \sum_{i=k}^l 1(\mathbf{U}_i^{k:l} \leq \mathbf{u}), \quad \mathbf{u} \in [0, 1]^d, \quad (6.8)$$

is the empirical copula of $\mathbf{X}_k, \dots, \mathbf{X}_l$ with the convention that $C_{k:l} = 0$ if $l < k$.

- In (6.8), the sample of pseudo-observations $\mathbf{U}_k^{k:l}, \dots, \mathbf{U}_l^{k:l}$ is defined by

$$\mathbf{U}_i^{k:l} = (F_{k:l,1}(X_{i1}), \dots, F_{k:l,d}(X_{id})) \frac{l-k+1}{l-k+2}, \quad i \in \{k, \dots, l\},$$

where $F_{k:l,j}$ is the empirical df of X_{kj}, \dots, X_{lj} .

- Similar to (6.3) and (6.4), a natural test statistic is then

$$\begin{aligned} S_n^C &= \sup_{t \in [0,1]} \int_{[0,1]^d} (\mathbb{D}_n^C(t, \mathbf{u}))^2 dC_{1:n}(\mathbf{u}) \\ &= \max_{1 \leq k \leq n-1} \frac{1}{n} \sum_{i=1}^n (\mathbb{D}_n^C(k/n, \mathbf{U}_i^{1:n}))^2. \end{aligned}$$

- The test based on S_n^C also relies on resampling.
- Conditions under which it is asymptotically valid under stationarity are stated in Bücher et al. (2014).

Example 6.2.2 (Test of stationarity based on S_n^C)

- We first apply the test based on S_n^C to the `rdj` data set:

```
1 > set.seed(314)
2 > withTime(cpCopula(Xrdj, b = NULL, method = "nonseq"))
```

```
Test for change-point detection sensitive to changes in the
copula with 'method'='nonseq'
```

```
data: Xrdj
cvmmmax = 57.28, p-value = 0.01049
```

```
User time: 51.5 sec
```

- The default value for the argument `method` is "`seq`", which leads to better finite-sample behavior for smaller samples.
- We chose "`nonseq`" here to speed-up the computations; more details can be obtained from `?cpCopula`.
- Similar to the result of S_n^H , the small p-value again provides evidence against stationarity for the `rdj` data set.

- For the bivariate daily log-returns obtained from the `gasoil` data set, we obtain:

```
1 > set.seed(137)
2 > withTime(cpCopula(Xgasoil, b = NULL, method = "nonseq"))
```

```
Test for change-point detection sensitive to changes in the
copula with 'method'='nonseq'
```

```
data: Xgasoil
cvmmmax = 6.2836, p-value = 0.6429
```

```
User time: 5.9 sec
```

- As measured by S_n^H and S_n^C , there is thus no evidence against stationarity in the bivariate log-returns computed from `gasoil`.



- Non-stationarity can also be a consequence of changes in the serial dependence of a multivariate time series.

- By construction, the tests based on S_n^H and S_n^C described previously are not designed to be sensitive to such changes.
- As demonstrated in Bücher et al. (2017), the ideas underlying the test based on S_n^C can, however, be easily adapted to derive a test sensitive to changes in the serial dependence of a univariate time series.
- Given a sequence of random variables $X_1, \dots, X_{n'}$ and an integer $p > 1$, the approach first consists of forming the $n = n' - p + 1$ p -dimensional vectors of observations

$$\mathbf{Y}_i = (X_i, \dots, X_{i+p-1}), \quad i \in \{1, \dots, n\}, \quad (6.9)$$

where $p - 1$ is the maximum lag under investigation and p is known as the *embedding dimension*.

- If $X_1, \dots, X_{n'}$ is a stretch from a continuous stationary time series, all the \mathbf{Y}_i have the same df, denoted by H^s .

- By Sklar's Theorem, H^s can be uniquely represented using a p -dimensional copula C^s such that

$$H^s(\mathbf{x}) = C^s(G(x_1), \dots, G(x_p)), \quad \mathbf{x} \in \mathbb{R}^p, \quad (6.10)$$

where G is the common df of each X_i .

- The copula C^s controls the dependence between the components of the \mathbf{Y}_i .
- Equivalently, it controls the serial dependence up to lag $p - 1$ in the time series, which is why it is sometimes called the lag $p - 1$ *autocopula* or *serial copula*.
- To derive tests of stationarity for continuous univariate time series particularly sensitive to changes in the serial dependence, Bücher et al. (2017) consider the following variant of the process \mathbb{D}_n^C in (6.7):

$$\begin{aligned} \mathbb{D}_n^{C^s}(t, \mathbf{u}) &= \sqrt{n} \lambda_n(0, t) \lambda_n(t, 1) (C_{1:\lfloor nt \rfloor}^s(\mathbf{u}) - C_{(\lfloor nt \rfloor + 1):n}^s(\mathbf{u})), \\ &\quad (t, \mathbf{u}) \in [0, 1]^{p+1}, \end{aligned}$$

where, for any $1 \leq k \leq l \leq n$,

$$C_{k:l}^s(\mathbf{u}) = \frac{1}{l-k+1} \sum_{i=k}^l \prod_{j=1}^p 1(G_{k:l}(X_{i+j-1}) \leq u_j), \quad \mathbf{u} \in [0, 1]^p, \quad (6.11)$$

and

$$G_{k:l}(x) = \frac{1}{l+p-k} \sum_{j=k}^{l+p-1} 1(X_j \leq x), \quad x \in \mathbb{R}. \quad (6.12)$$

- The quantity $C_{k:l}^s$ is a non-parametric estimator of C^s based on $\mathbf{Y}_k, \dots, \mathbf{Y}_l$ that could be called the lag $p - 1$ *empirical autocopula*.
- The latter was used, for instance, in Genest and Rémillard (2004) for testing serial independence as we shall see in Section 6.2.2.
- It can be verified that it is a straightforward transposition (to the serial context under consideration) of the definition adopted in (6.8).

- A natural **test statistic** is then

$$\begin{aligned} S_n^{C^s} &= \sup_{t \in [0,1]} \int_{[0,1]^p} (\mathbb{D}_n^{C^s}(t, \mathbf{u}))^2 dC_{1:n}^s(\mathbf{u}) \\ &= \max_{1 \leq k \leq n-1} \int_{[0,1]^p} (\mathbb{D}_n^{C^s}(k/n, \mathbf{u}))^2 dC_{1:n}^s(\mathbf{u}). \end{aligned}$$

- Similar to the tests based on S_n^H and S_n^C , the computation of an approximate p-value for the test based on $S_n^{C^s}$ relies on **resampling**.
- The **asymptotic validity** of the resampling scheme **under stationarity** is shown in Bücher et al. (2017).
- Prior to applying the test based on $S_n^{C^s}$ to a univariate time series, it is necessary to **choose the embedding dimension p** or, equivalently, the maximum lag $p - 1$.
- This aspect is discussed in Bücher et al. (2017, Section 3) where a first rough recommendation is to consider $p \in \{2, 3, 4\}$.

- Multivariate extensions of the test based on $S_n^{C^s}$ are mentioned in Bücher et al. (2017) but seem to need more investigation, in particular from the point of view of their finite-sample properties.
- In the presence of a multivariate time series, an **imperfect alternative** is to **apply** the test based on $S_n^{C^s}$ **to each component series**.

Example 6.2.3 (Test of stationarity based on $S_n^{C^s}$)

- We first apply the test based on $S_n^{C^s}$ to the first component series of the **rdj** data set with $p - 1 = 1$:

```
1 > set.seed(3355)
2 > withTime(cpAutocop(Xrdj[,1], lag = 1))
```

Test of change-point detection sensitive to changes in the
2-dimensional autocopula

```
data: Xrdj[, 1]
cvmmax = 37.553, p-value = 0.03447
```

User time: 18.9 sec

- The p-value provides **weak evidence against stationarity** for the rdj data set.
- If the **maximum lag** to be investigated is set to **2**, we obtain:

```
1 > set.seed(3355)
2 > withTime(cpAutocop(Xrdj[,1], lag = 2))
```

```
Test of change-point detection sensitive to changes in the
3-dimensional autocopula
```

```
data: Xrdj[, 1]
cvmmax = 35.028, p-value = 0.04645
```

```
User time: 18.5 sec
```

Finally, with $p - 1 = 3$, we get:

```
1 > set.seed(3355)
2 > withTime(cpAutocop(Xrdj[,1], lag = 3))
```

```
Test of change-point detection sensitive to changes in the
4-dimensional autocopula
```

```
data: Xrdj[, 1]
cvmmmax = 23.419, p-value = 0.08242

User time: 23.9 sec
```

- The increase in the p-value as p is increased is due to the fact that, as discussed in Bücher et al. (2017), in many cases the power of the test based on $S_n^{C^s}$ decreases as p is increased.
- This is partly a consequence of the fact that the underlying empirical serial copulas are of dimension p .
- For the bivariate daily log-returns obtained from the `gasoil` data set, we obtain, for the **first component series** with $p - 1 = 1$:

```
1 > set.seed(3105)
2 > withTime(cpAutocop(Xgasoil[,1], lag = 1))
```

```
Test of change-point detection sensitive to changes in the
2-dimensional autocopula
```

```
data: Xgasoil[, 1]
cvmmmax = 11.309, p-value = 0.2303
```

User time: 4.4 sec

- For the second component series, we have with $p - 1 = 1$:

```
1 > set.seed(2895)
2 > withTime(cpAutocop(Xgasoil[,2], lag = 1))
```

```
Test of change-point detection sensitive to changes in the
2-dimensional autocopula
```

```
data: Xgasoil[, 2]
cvmmax = 13.243, p-value = 0.1643
```

User time: 4.4 sec

- The results are qualitatively similar when larger values of p are considered.
- Overall, there is thus no evidence against stationarity in the bivariate log-returns computed from gasoil.



- It is important to keep in mind that, although it is designed to be particularly sensitive to changes in the serial dependence, the test based on $S_n^{C^s}$ is a **test of stationarity**.
- Consequently, a **small p-value** should not be used to conclude that there is a **change in the serial dependence** unless one has reasons to believe that, for the data at hand, non-stationarity can only be a consequence of changes in the serial dependence.
- Similarly, a **small p-value** for the test based on S_n^H (respectively, S_n^C) **should not be used to conclude that \mathcal{H}_0** in (6.1) (respectively, $\mathcal{H}_{0,c}$ in (6.6)) **does not hold** unless one is ready to believe that, for the data at hand, non-stationarity can only be a consequence of departures from \mathcal{H}_0 (respectively, $\mathcal{H}_{0,c}$).

6.2.2 Tests of serial independence

- If there is no evidence against the hypothesis that X_1, \dots, X_n is a stretch from a stationary multivariate time series, a next natural question is whether X_1, \dots, X_n are serially independent.
- A classical approach to assess whether serial dependence is present in the data at hand is to estimate the autocorrelation and cross-correlation functions of the multivariate time series and complement this by Ljung–Box-type portmanteau tests.

Example 6.2.4 (Correlogram and Ljung–Box test of serial independence)

- We consider again the bivariate log-returns computed from the gasoil data set initially described in Example 6.2.1.
- When investigating the presence of serial dependence in financial log-returns, it is customary to consider the time series of squared (or absolute)

log-returns.

- This is due to a stylized fact known as *volatility clustering*, that is, the tendency for large changes to be followed by large changes, although not necessarily with the same sign; see McNeil et al. (2015, Section 3.1.1).
- The **correlogram** representing the estimated autocorrelation and cross-correlation functions of the squared log-returns can be displayed as follows:

```
1 > colnames(Xgasoil) <- c("Oil", "Gas")
2 > acf(Xgasoil^2, ci.col = 1)
```

- The resulting plots are shown in Figure 6.5.
- The correlogram of the second squared component series appears to display **significant autocorrelations**.
- To **formally test for serial independence**, we apply **Ljung–Box** tests (via `Box.test()`) to both squared component series.

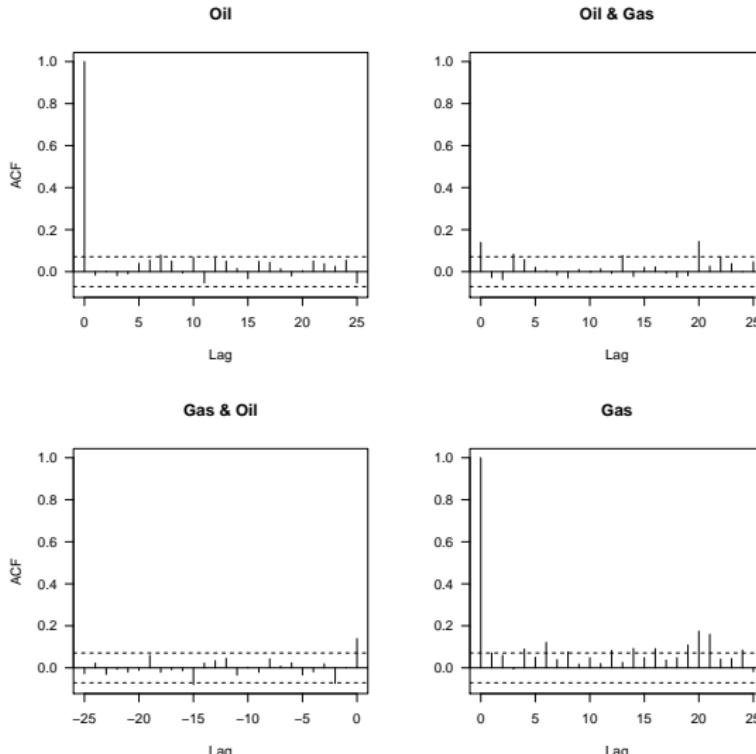


Figure 6.5 Correlogram of the bivariate log-returns of the gasoil data.

- We first test for significant autocorrelations up to lag 5, and then, up to lag 20:

```
1 > Box.test(Xgasoil[,1]^2, lag = 5, type = "Ljung-Box")
```

Box-Ljung test

```
data: Xgasoil[, 1]^2  
X-squared = 1.7878, df = 5, p-value = 0.8777
```

```
1 > Box.test(Xgasoil[,2]^2, lag = 5, type = "Ljung-Box")
```

Box-Ljung test

```
data: Xgasoil[, 2]^2  
X-squared = 14.752, df = 5, p-value = 0.01148
```

```
1 > Box.test(Xgasoil[,1]^2, lag = 20, type = "Ljung-Box")
```

Box-Ljung test

```
data: Xgasoil[, 1]^2  
X-squared = 26.712, df = 20, p-value = 0.1435
```

```
1 > Box.test(Xgasoil[,2]^2, lag = 20, type = "Ljung-Box")
```

```
Box-Ljung test
```

```
data: Xgasoil[, 2]^2  
X-squared = 90.924, df = 20, p-value = 5.108e-11
```

- The tests provide **strong evidence against serial independence** in the second (squared) component series, that is, the squared log-returns of gas prices.



- Portmanteau tests (such as the Ljung–Box test) applied on squared observations are, however, known to be often **too liberal** when the observations are **heavy-tailed**.
- More **robust** tests of serial independence, based on **ranks**, for instance, might be preferable.

- One possible such test is the one based on the [serial empirical copula](#) proposed by Genest and Rémillard (2004).
- It is a [serial extension](#) of the [multivariate test of independence](#) initially proposed by Deheuvels (1981) and briefly presented in Section 5.2.1.
- Given a [stationary](#) sequence of continuous random variables $X_1, \dots, X_{n'}$ and an [embedding dimension](#) $p > 1$, the approach of Genest and Rémillard (2004) consists of first forming the $n = n' - p + 1$ [p-dimensional random vectors](#) \mathbf{Y}_i , $i \in \{1, \dots, n\}$, defined by (6.9).
- It follows from the stationarity assumption that [all the \$\mathbf{Y}_i\$](#) have the same [df](#) H^s which, as mentioned in the previous section, can be [decomposed](#) as in (6.10) in terms of the [serial copula](#) C^s and the common [df](#) G of each X_i .
- Starting from the fact that, [under serial independence](#) of $X_1, \dots, X_{n'}$, C^s is the [independence copula](#) Π , Genest and Rémillard (2004) consider

test statistics derived from the empirical process

$$\sqrt{n}(C_{1:n}^s(\mathbf{u}) - \Pi(\mathbf{u})), \quad \mathbf{u} \in [0, 1]^p,$$

where $C_{1:n}^s$ is the serial empirical copula defined by (6.11).

- This amounts to working with the **ranks** $R_1, \dots, R_{n'}$ associated with $X_1, \dots, X_{n'}$ as $R_i = n'G_{1:n}(X_i)$, $i \in \{1, \dots, n'\}$, where $G_{1:n}$ is defined by (6.12).
- A sensible **test statistic** is then

$$S_n^{\Pi^s} = n \int_{[0,1]^p} (C_{1:n}^s(\mathbf{u}) - \Pi(\mathbf{u}))^2 d\mathbf{u}.$$

- Its **realizations under serial independence** can be obtained by **simulation**, thereby allowing one to compute an **approximate p-value** for $S_n^{\Pi^s}$ **empirically**.
- The previous construction was extended to multivariate time series by Kojadinovic and Yan (2011b) who used a permutation principle to compute approximate p-values.

Example 6.2.5 (Tests of serial independence based on $S_n^{\Pi^s}$)

- We consider again the bivariate log-returns of the `gasoil` data, which, according to Examples 6.2.1, 6.2.2 and 6.2.3, might well be a stretch from a stationary time series.
- We choose to apply the test based on $S_n^{\Pi^s}$ with $p = 6$, which corresponds to a maximum lag of 5.
- To compute an approximate p-value for $S_n^{\Pi^s}$, it is first necessary to generate a large number, say 1000, of realizations of the test statistic under serial independence for sample size n and embedding dimension p :

```
1 > set.seed(137)
2 > sI.d <- withTime(serialIndepTestSim(nrow(Xgasoil), lag.max = 5))
3 > sI.d$sys.time # the run time
```

User time: 321.8 sec

- The generated realizations of $S_n^{\Pi^s}$, stored in the output `sI.d$value` of the function `serialIndepTestSim()`, can then be used to carry out the test on the two squared component series:

```
1 > serialIndepTest(Xgasoil[,1]^2, d = sI.d$value)
```

```
Global Cramer-von Mises statistic: 0.03519808 with p-value 0.02247752
Combined p-values from the Mobius decomposition:
  0.2862138 from Fisher's rule,
  0.3361638 from Tippett's rule.
```

```
1 > serialIndepTest(Xgasoil[,2]^2, d = sI.d$value)
```

```
Global Cramer-von Mises statistic: 0.07246049 with p-value 0.002497502
Combined p-values from the Mobius decomposition:
  0.08841159 from Fisher's rule,
  0.0004995005 from Tippett's rule.
```

- The first p-values in the outputs are those corresponding to the test statistic $S_n^{\text{II}^s}$.
- The definitions of the test statistics corresponding to the remaining two p-values can be found in Genest and Rémillard (2004).

- The results are in line with those obtained from the Ljung–Box test in Example 6.2.4 in that there is strong evidence against serial independence in the second component series.



6.2.3 Models for multivariate time series based on conditional copulas

- Suppose that there is no evidence against stationarity but evidence against serial independence of $\mathbf{X}_1, \dots, \mathbf{X}_n$, implying that $\mathbf{X}_1, \dots, \mathbf{X}_n$ can be regarded as a stretch from a stationary d -dimensional time series.
- As explained at the beginning of Section 6.2, in such a time series context and with forecasting in mind, the interest often does not lie in the (unconditional) distribution of the random vector \mathbf{X} of which $\mathbf{X}_1, \dots, \mathbf{X}_n$ are serially dependent copies, but rather in the conditional

distribution of \mathbf{X}_i given the sigma-field \mathcal{G}_{i-1} generated by $\mathbf{X}_1, \dots, \mathbf{X}_{i-1}$, that is, the information available at time $i - 1$.

- Under additional assumptions, the conditional df of \mathbf{X}_i given \mathcal{G}_{i-1} can be decomposed as in Sklar's Theorem which leads to the concept of *conditional copula* initially introduced in Patton (2006); see also Fermanian and Wegkamp (2012).
- The aim of this section is to provide a brief overview of time series models based on this notion.
- To introduce the concept of conditional copula, we first consider a simpler setting: let (\mathbf{X}, \mathbf{Z}) be a $(d + q)$ -dimensional random vector, let $\text{ran } \mathbf{Z} = \{\mathbf{z} \in \mathbb{R}^q : \mathbb{P}(\mathbf{Z} \in (\mathbf{z} - \mathbf{h}, \mathbf{z})) > 0 \text{ for all } \mathbf{h} > \mathbf{0}\}$ be the range of the random vector \mathbf{Z} and, given $\mathbf{z} \in \text{ran } \mathbf{Z}$, let us focus on the conditional df of \mathbf{X} given $\mathbf{Z} = \mathbf{z}$, that is, on the conditional df

$$H_{\mathbf{z}}(\mathbf{x}) = \mathbb{P}(X_1 \leq x_1, \dots, X_d \leq x_d \mid \mathbf{Z} = \mathbf{z}), \quad \mathbf{x} \in \mathbb{R}^d. \quad (6.13)$$

- Assuming that H_z is continuous, we immediately obtain from Sklar's Theorem that

$$H_z(\mathbf{x}) = C_z(F_{z,1}(x_1), \dots, F_{z,d}(x_d)), \quad \mathbf{x} \in \mathbb{R}^d, \quad (6.14)$$

where $F_{z,j}(x) = \mathbb{P}(X_j \leq x \mid \mathbf{Z} = z)$, $j \in \{1, \dots, d\}$, are the univariate margins of H_z and C_z is the copula of H_z .

- Equivalently,

$$C_z(\mathbf{u}) = \mathbb{P}(F_{z,1}(X_1) \leq u_1, \dots, F_{z,d}(X_d) \leq u_d \mid \mathbf{Z} = z), \quad \mathbf{u} \in [0, 1]^d. \quad (6.15)$$

- Assuming that H_z is continuous for all $z \in \text{ran } \mathbf{Z}$ (which implies that (6.14) holds for all $z \in \text{ran } \mathbf{Z}$), the underlying conditional copula is the random function C_Z obtained by replacing z by (an independent copy of) \mathbf{Z} in (6.15).
- Providing a rigorous definition of the notion of conditional copula in the time series setting considered in this section is mathematically more involved; see Fermanian and Wegkamp (2012).

- We shall only attempt to convey the **main intuitions**.
- Recall that we are interested in modeling the **conditional df** of X_i given the information \mathcal{G}_{i-1} at time $i - 1$, that is, the conditional df

$$H_{\mathcal{G}_{i-1}}(\mathbf{x}) = \mathbb{P}(X_{i1} \leq x_1, \dots, X_{id} \leq x_d \mid \mathcal{G}_{i-1}), \quad \mathbf{x} \in \mathbb{R}^d. \quad (6.16)$$

- Furthermore, let $F_{\mathcal{G}_{i-1},j}(x) = \mathbb{P}(X_{ij} \leq x \mid \mathcal{G}_{i-1})$, $x \in \mathbb{R}$, $j \in \{1, \dots, d\}$, denote the corresponding margins.
- Then, under continuity and measurability assumptions, there exists an almost surely unique random function $C_{\mathcal{G}_{i-1}}$, referred to as a **conditional copula**, such that

$$H_{\mathcal{G}_{i-1}}(\mathbf{x}) = C_{\mathcal{G}_{i-1}}(F_{\mathcal{G}_{i-1},1}(x_1), \dots, F_{\mathcal{G}_{i-1},d}(x_d)), \quad \mathbf{x} \in \mathbb{R}^d.$$

- Starting from the above **decomposition**, a first natural approach to model $H_{\mathcal{G}_{i-1}}$ is to consider **parametric families** for the **marginal dfs** $F_{\mathcal{G}_{i-1},j}$, $j \in \{1, \dots, d\}$, and the **conditional copula** $C_{\mathcal{G}_{i-1}}$, and to **allow the**

corresponding parameter vectors to change according to the information \mathcal{G}_{i-1} available at time $i - 1$.

- The latter thus incurs an additional modeling difficulty compared to, for instance, the (unconditional) setting considered in Section 4.1.1 since it is necessary to specify how the marginal and the copula parameters change with \mathcal{G}_{i-1} .
- When dealing with financial time series, the most frequently encountered model in this setting, see Patton (2006), Patton (2012), Patton (2013) and the references therein, is as follows:
 - ▶ For any $j \in \{1, \dots, d\}$, it is assumed that

$$X_{ij} = \mu_{ij}(\beta_j) + \sigma_{ij}(\beta_j)\varepsilon_{ij}, \quad (6.17)$$

where $\mu_{ij}(\beta_j) = \mathbb{E}(X_{ij} | \mathcal{G}_{i-1})$ is the conditional mean of X_{ij} given \mathcal{G}_{i-1} , $\sigma_{ij}^2(\beta_j) = \text{Var}(X_{ij} | \mathcal{G}_{i-1})$ is the conditional variance of X_{ij} given \mathcal{G}_{i-1} and the conditional distribution of the innovations ε_{ij} given \mathcal{G}_{i-1} does not depend on \mathcal{G}_{i-1} .

- ▶ In addition, the conditional mean and variance in (6.17) are defined up to the finite-dimensional parameter vector β_j .
- ▶ For identifiability, it is further assumed that, for any $j \in \{1, \dots, d\}$, the (non-random) conditional distribution of ε_{ij} given \mathcal{G}_{i-1} has mean zero and variance one and that its df belongs to an absolutely continuous parametric family of univariate dfs $\mathcal{F}_j = \{F_{j,\gamma_j} : \gamma_j \in \Gamma_j\}$, where Γ_j is the parameter space.
- ▶ Note that the conditional distribution of $\varepsilon_i = (\varepsilon_{i1}, \dots, \varepsilon_{id})$ given \mathcal{G}_{i-1} may depend on \mathcal{G}_{i-1} . In other words, although the univariate margins of the conditional df of ε_i given \mathcal{G}_{i-1} are not random (they do not depend on \mathcal{G}_{i-1}), the conditional copula of ε_i given \mathcal{G}_{i-1} may depend on \mathcal{G}_{i-1} .
- ▶ It is additionally assumed that there exists an absolutely continuous parametric family of copulas $\mathcal{C} = \{C_\theta : \theta \in \Theta\}$, where Θ is the parameter space, and a parametric *copula calibration function* φ that

transforms \mathcal{G}_{i-1} into $\theta_{i-1} \in \Theta$ so that the copula of the conditional distribution of ε_i given \mathcal{G}_{i-1} is $C_{\varphi(\mathcal{G}_{i-1})} = C_{\theta_{i-1}}$.

- ▶ The calibration function φ governs the **temporal dynamics** of the parameter vector of the copula: it determines how the copula parameter vector changes with the information \mathcal{G}_{i-1} available at time $i - 1$.
- ▶ The **copula calibration function** φ is further assumed to be **completely defined up to a finite-dimensional parameter vector β** (which is not to be confused with the finite-dimensional parameter vectors β_1, \dots, β_d defining the conditional means and variances in (6.17)).
- As a consequence, under this model, the **conditional df of ε_i given \mathcal{G}_{i-1}** can be written as $C_{\varphi(\mathcal{G}_{i-1})}(F_{1,\gamma_1}(\cdot), \dots, F_{d,\gamma_d}(\cdot))$.
- It follows from (6.17) that the conditional df $H_{\mathcal{G}_{i-1}}$ of X_i given \mathcal{G}_{i-1} ,

see (6.16), can be expressed, for $\boldsymbol{x} \in \mathbb{R}^d$, as

$$H_{\mathcal{G}_{i-1}}(\boldsymbol{x}) = C_{\varphi(\mathcal{G}_{i-1})} \left(F_{1,\gamma_1} \left(\frac{x_1 - \mu_{i1}(\boldsymbol{\beta}_1)}{\sigma_{i1}(\boldsymbol{\beta}_1)} \right), \dots, F_{d,\gamma_d} \left(\frac{x_d - \mu_{id}(\boldsymbol{\beta}_d)}{\sigma_{id}(\boldsymbol{\beta}_d)} \right) \right).$$

- Denoting by c_{θ} the density of C_{θ} and by f_{j,γ_j} the density of F_{j,γ_j} , $j \in \{1, \dots, d\}$, we immediately obtain that the **conditional density** $h_{\mathcal{G}_{i-1}}$ of \boldsymbol{X}_i given \mathcal{G}_{i-1} can be written as

$$h_{\mathcal{G}_{i-1}}(\boldsymbol{x}) = c_{\varphi(\mathcal{G}_{i-1})} \left(F_{1,\gamma_1} \left(\frac{x_1 - \mu_{i1}(\boldsymbol{\beta}_1)}{\sigma_{i1}(\boldsymbol{\beta}_1)} \right), \dots, F_{d,\gamma_d} \left(\frac{x_d - \mu_{id}(\boldsymbol{\beta}_d)}{\sigma_{id}(\boldsymbol{\beta}_d)} \right) \right) \\ \cdot \prod_{j=1}^d \frac{1}{\sigma_{ij}(\boldsymbol{\beta}_j)} f_{j,\gamma_j} \left(\frac{x_j - \mu_{ij}(\boldsymbol{\beta}_j)}{\sigma_{ij}(\boldsymbol{\beta}_j)} \right).$$

- The **log-likelihood** of the model is then

$$\ell_n(\boldsymbol{\beta}, \boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_d, \boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_d) = \log \prod_{i=1}^n h_{\mathcal{G}_{i-1}}(\boldsymbol{X}_i) = \sum_{i=1}^n \log h_{\mathcal{G}_{i-1}}(\boldsymbol{X}_i).$$

- The latter is often maximized in stages in the spirit of the inference functions for margins estimator presented in Section 4.1.1.
- The marginal parameters β_j and γ_j , $j \in \{1, \dots, d\}$, are estimated first in d separate steps, each involving one of the d component series.
- Denoting by $\beta_{j,n}$ and $\gamma_{j,n}$ the estimators of β_j and γ_j , respectively, the estimated standardized residuals are given by

$$\varepsilon_{ij,n} = \frac{X_{ij} - \mu_{ij}(\beta_{j,n})}{\sigma_{ij}(\beta_{j,n})}, \quad i \in \{1, \dots, n\}, \quad j \in \{1, \dots, d\}. \quad (6.18)$$

- The parameter β specifying the copula calibration function can then be estimated by

$$\underset{\beta}{\operatorname{argmax}} \sum_{i=1}^n \log c_{\varphi(\mathcal{G}_{i-1})}(F_{1,\gamma_{1,n}}(\varepsilon_{i1,n}), \dots, F_{d,\gamma_{d,n}}(\varepsilon_{id,n})).$$

- A simplified version of the above model is investigated in Chen and Fan (2006) and Rémillard (2017), among others: the main difference is that

the conditional copula of ε_i given \mathcal{G}_{i-1} is also assumed not to depend on \mathcal{G}_{i-1} , implying that the conditional distribution of ε_i given \mathcal{G}_{i-1} does not depend on \mathcal{G}_{i-1} , or equivalently, that the copula calibration function φ is constant.

- As a consequence, its parameter vector β can be identified with the unknown parameter vector θ of the underlying non-time-varying copula and $\varepsilon_1, \dots, \varepsilon_n$ are iid.
- Chen and Fan (2006) then propose to estimate β semi-parametrically in the spirit of the maximum pseudo-likelihood estimator presented in Section 4.1.2.
- The resulting estimator β_n of $\beta = \theta$ is given by

$$\beta_n = \operatorname{arg\!sup}_{\theta \in \Theta} \sum_{i=1}^n \log c_\theta(F_{1,n}(\varepsilon_{i1,n}), \dots, F_{d,n}(\varepsilon_{id,n})), \quad (6.19)$$

where, for any $j \in \{1, \dots, d\}$,

$$F_{j,n}(x) = \frac{1}{n+1} \sum_{i=1}^n 1(\varepsilon_{ij,n} \leq x), \quad x \in \mathbb{R}. \quad (6.20)$$

- The estimator β_n is thus nothing else but the maximum pseudo-likelihood estimator of Section 4.1.2 computed from the estimated standardized residuals given in (6.18).
- Under correct marginal specifications, Chen and Fan (2006) show the somewhat surprising result that the above estimator behaves asymptotically as if it were based on the unobservable innovations $\varepsilon_i = (\varepsilon_{i1}, \dots, \varepsilon_{id})$, $i \in \{1, \dots, n\}$, and not on the estimated standardized residuals $\varepsilon_{i,n} = (\varepsilon_{i1,n}, \dots, \varepsilon_{id,n})$, $i \in \{1, \dots, n\}$, defined by (6.18).
- Under similar assumptions, Rémillard (2017) shows that the empirical copula process (see Section 4.2.1) based on the estimated standardized residuals has the same asymptotics as if it were based on the unobservable innovations instead.

- In other words, the aforementioned asymptotics do not depend on the conditional mean and variance parameters.
- Hence, as explained in Rémillard (2017), provided the assumptions of the aforementioned simplified model hold, one can apply the rank-based copula inference procedures of Chapters 4 and 5 on the estimated standardized residuals as if they were the innovations.
- In applications, a frequent choice for the marginal models are ARMA models for the conditional means $\mu_{ij}(\beta_j)$, $j \in \{1, \dots, d\}$, and GARCH models for the conditional variances $\sigma_{ij}^2(\beta_j)$, $j \in \{1, \dots, d\}$.
- In a related way, Chen and Fan (2006) and Jondeau and Rockinger (2006) seem to be among the first to popularize the expression *copula–GARCH* to refer to certain models of this type.

Example 6.2.6 (Conditional modeling based on ARMA–GARCH)

- From Examples 6.2.1, 6.2.2 and 6.2.3, we have no reasons to believe that the bivariate log-returns computed from the `gasoil` data set do

not form a sample from a stationary multivariate time series.

- However, as seen in Example 6.2.5, the log-returns cannot be considered iid.
- To apply the aforementioned modeling steps, we consider ARMA–GARCH models for the margins using their implementation in the R package rugarch of Ghalanos (2017).
- For the sake of simplicity, we do not attempt to find the best fitting ARMA–GARCH models but simply consider the popular ARMA(1,1)–GARCH(1,1) model with t innovations (scaled to have variance one) for both component series.
- Furthermore, after finding no evidence of time varying dependence using the test based on S_n^C presented in Section 6.2.1, we decide to link the marginal innovations with a t copula.
- Note that a more adequate modeling of these data is carried out in Grégoire et al. (2008).

- For illustration purposes, we also simulate 1000 paths of size 260 (the latter are roughly the number of trading days in one year) from the fitted bivariate copula–GARCH time-series model for the log-returns.
- We then use these paths to estimate the one-year ahead bivariate predictive density of oil and gas prices.
- Model specification is carried out as follows based on the function ugarchspec() of rugarch:

```

1 > library(rugarch)
2 > ## Specify ARMA(1,1)-GARCH(1,1) model with Student t innovations
3 > meanModel <- list(armaOrder = c(1,1)) # ARMA(1,1)
4 > varModel  <- list(model = "sGARCH", garchOrder = c(1,1)) # GARCH(1,1)
5 > uspec <- ugarchspec(varModel, mean.model = meanModel,
6                               distribution.model = "std") # scaled Student t

```

- Next, we estimate the component models with the function ugarchfit() and extract the estimated standardized residuals in (6.18):

```

1 > ## Fit marginal ARMA-GARCH models
2 > fit <- apply(Xgasoil, 2, function(x) ugarchfit(uspec, data = x))
3 > ## Extract the estimated standardized residuals

```

```
4 > eps <- sapply(fit, residuals, standardize = TRUE) # standardized residuals  
5 > (nus <- sapply(fit, function(x) x@fit$coef[["shape"]])) # fitted d.o.f.
```

Oil	Gas
13.298668	5.821641

- Assuming correct marginal specifications, the function `cpCopula()` of the R package `npcp` can be used to test the hypothesis that the conditional copula does not depend on \mathcal{G}_{i-1} , or, equivalently, that it is not time-varying.
- Recall from Section 6.2.1 that the computation of an approximate p-value for the underlying test statistic S_n^C requires the knowledge of a bandwidth parameter corresponding to the argument `b`.
- Since the estimated standardized residuals should be approximately serially independent, we change the default `b = NULL` (forcing the estimation of the bandwidth parameter) to `b = 1` (see `?cpCopula` for more details):

```
1 > set.seed(2013)
2 > withTime(cpCopula(eps, b = 1, method = "nonseq"))
```

```
Test for change-point detection sensitive to changes in the
copula with 'method'='nonseq'
```

```
data: eps
cvmmax = 6.6343, p-value = 0.5839
```

```
User time: 5.5 sec
```

- The large p-value provides no evidence against a constant, that is, non-time-varying, conditional copula.
- We then form the corresponding pseudo-observations based on (6.20):

```
1 > U <- pobs(eps) # pseudo-observations from the residuals eps
2 > plot(U, xlab = expression(U[1]), ylab = expression(U[2]))
```

- The left-hand side of Figure 6.6 shows a scatter plot of the pseudo-observations of the estimated standardized residuals. Next, we fit a *t* copula to these pseudo-observations by means of (6.19):

```
1 > ## Fit a t copula to the estimated standardized residuals  
2 > fitcop <- fitCopula(tCopula(), data = U, method = "mpl")  
3 > fitcop@estimate # estimated correlation parameter rho and d.o.f. nu
```

```
[1] 0.5019015 45.5268659
```

```
1 > cop <- fitcop@copula # fitted t copula
```

- Note the large degrees of freedom, so we could have equally well chosen a [normal copula](#) here.
- We are now in the position to [simulate the cross-sectionally dependent innovations](#) $\varepsilon_i = (\varepsilon_{i1}, \varepsilon_{i2})$ of the two component time-series models.
- Specifically, to estimate the [bivariate predictive density of oil and gas prices one year ahead](#), we simulate 1000 paths of size 260 of oil and gas prices via `ugarchsim()`:

```
1 > ## Simulate from the bivariate model  
2 > ## 1) Simulate from the fitted copula  
3 > set.seed(271) # set seed  
4 > n.sim <- 260 # sample size
```

```

5 > m.sim <- 1000 # number of paths
6 > U. <- rCopula(n.sim * m.sim, cop) # simulate from the fitted copula
7 > ## 2) Quantile-transform the corresponding innovations
8 > ## Note: eps have to be standardized (mean 0, variance 1) for ugarchsim()
9 > eps. <- sapply(1:2, function(j)
10   sqrt((nus[j]-2)/nus[j]) * qt(U.[,j], df = nus[j]))
11 > ## 3) Feed the (cross-sec. dependent) innovations to the marginal ARMA-GARCH
12 > ## models and simulate from them
13 > sim <- lapply(1:2, function(j)
14   ugarchsim(fit[[j]], # fitted marginal ARMA-GARCH model
15     n.sim = n.sim, # sample size
16     m.sim = m.sim, # number of trajectories/paths
17     custom.dist = list(name = "sample", # our innovations
18       distfit = matrix(eps.[,j], ncol = m.sim))))
19 > ## 4) Extract the simulated (cross-sec. dependent) series X_t and build the
20 > ## corresponding (predicted/simulated) oil and gas prices
21 > X. <- lapply(sim, function(x) fitted(x)) # equal to seriesSim in x@simulation
22 > S.t <- as.numeric(tail(gasoil, n = 1)[,2:3]) # last available prices S_t
23 > library(qrmtools) # for returns()
24 > S. <- lapply(1:2, function(j) # predicted prices for each stock
25   returns(X.[[j]], inverse = TRUE, start = rep(S.t[j], m.sim)))
26 > S.T <- sapply(1:2, function(j) tail(S.[[j]], n = 1)) # pick out prices at T

```

- Based on these 1000 sample paths, we can then compute an estimate of the bivariate predictive density for the daily oil and gas prices for the

following year:

```
1 > library(MASS)
2 > pred.dens <- kde2d(S.T[,1], S.T[,2], n = 300, lims = c(0, 200, 0, 15))
3 > image(pred.dens, xlab = "Oil price", ylab = "Gas price",
4           col = gray(seq(1, 0, length.out = 100)))
```

- An estimate of the predictive density is displayed on the right-hand side of Figure 6.6; note that the larger the density, the darker the shade of gray.



- Additional examples of marginal conditional modeling based on ARMA–GARCH models prior to copula inference can be found, for instance, in the documentation of the function `fit_ARMA_GARCH()` of the R package `qrmttools`; see also the vignette `copula_GARCH` of `copula`.
- Other types of time series models based on copulas are, for instance, discussed in Darsow et al. (1992), Ibragimov (2009), Beare (2010) and Rémillard et al. (2012); see also the references therein.

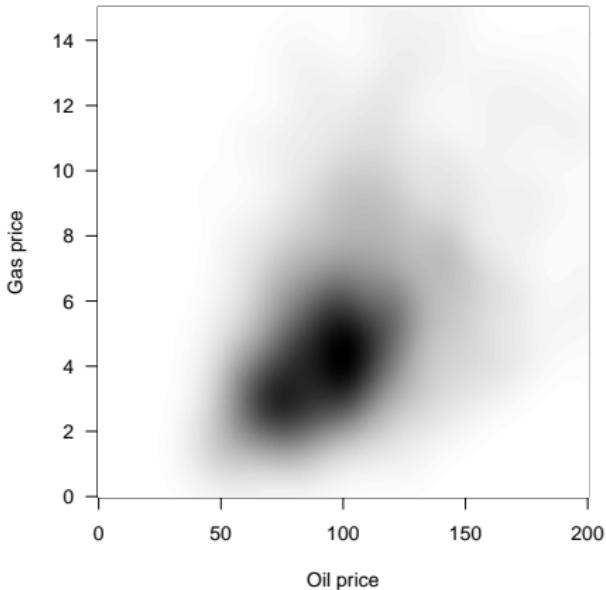
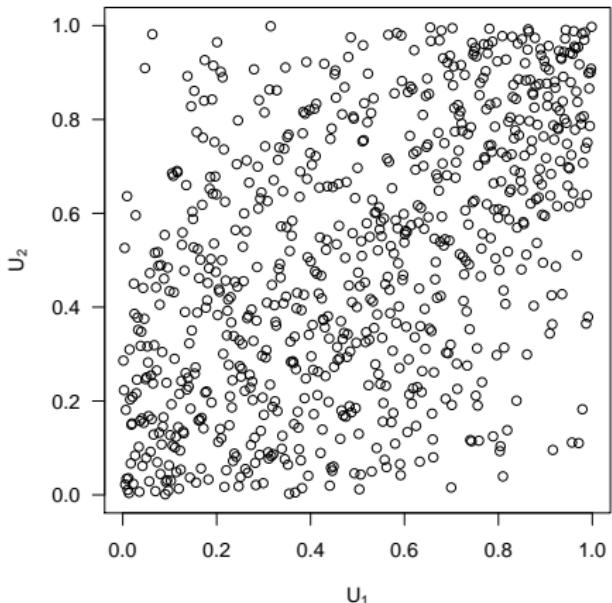


Figure 6.6 Pseudo-observations of the residuals obtained after ARMA–GARCH modeling of the oil and gas log-returns (left) and estimated predictive density for oil and gas prices one year ahead (right).

6.3 Regression

- As already mentioned in Section 6.2.3, in many applications, besides the random vector \mathbf{X} of interest, one also observes a q -variate random vector \mathbf{Z} of covariates or explanatory variables.
- The focus then typically shifts from modeling the unconditional distribution of \mathbf{X} to modeling the conditional distribution of \mathbf{X} given the covariates \mathbf{Z} .
- Under continuity assumptions, the latter can be based, again, on the concept of conditional copula already introduced in Section 6.2.3.

References

- Aue, A. and Horváth, L. (2013), Structural breaks in time series, *Journal of Time Series Analysis*, 34(1), 1–16.
- Beare, B. K. (2010), Copulas and temporal dependence, *Econometrica*, 78(1), 395–410.
- Ben Ghorbal, M., Genest, C., and Nešlehová, J. G. (2009), On the test of Ghoudi, Khoudraji, and Rivest for extreme-value dependence, *The Canadian Journal of Statistics*, 37(4), 534–552.
- Berg, D. (2009), Copula goodness-of-fit testing: An overview and power comparison, *The European Journal of Finance*, 15, 675–701.
- Bücher, A. and Kojadinovic, I. (2015), An overview of nonparametric tests of extreme-value dependence and of some related statistical procedures, *Extreme Value Modeling and Risk Analysis: Methods and Applications*, ed. by D. Dey and J. Yan, Boca Raton, Florida: Chapman & Hall/CRC, 377–398.

- Bücher, A. and Kojadinovic, I. (2016), A dependent multiplier bootstrap for the sequential empirical copula process under strong mixing, *Bernoulli*, 22(2), 927–968.
- Bücher, A., Kojadinovic, I., Rohmer, T., and Segers, J. (2014), Detecting changes in cross-sectional dependence in multivariate time series, *Journal of Multivariate Analysis*, 132, 111–128.
- Bücher, A., Fermanian, J.-D., and Kojadinovic, I. (2017), Combining cumulative sum change-point detection tests for assessing the stationarity of continuous univariate time series, tech. rep., ArXiv E-prints, arXiv: 1709.02673.
- Cardinali, A. and Nason, G. P. (2013), Costationarity of locally stationary time series using costat, *Journal of Statistical Software*, 55(1), 1–22.
- Chen, X. and Fan, Y. (2005), Pseudo-likelihood ratio tests for semiparametric multivariate copula model selection, *The Canadian Journal of Statistics*, 33, 389–414.

- Chen, X. and Fan, Y. (2006), Estimation and model selection of semiparametric copula-based multivariate dynamic models under copula misspecification, *Journal of Econometrics*, 135, 125–154.
- Chen, X., Fan, Y., and Tsyrennikov, V. (2006), Efficient estimation of semiparametric multivariate copula models, *Journal of the American Statistical Association*, 101, 1228–1240.
- Cherubini, U., Luciano, E., and Vecchiato, W. (2004), Copula Methods in Finance, Chichester, UK: John Wiley & Sons.
- Cherubini, U., Mulinacci, S., Gobbi, F., and Romagnoli, S. (2011), Dynamic Copula Methods in Finance, Chichester, UK: John Wiley & Sons.
- Claeskens, G. and Hjort, N. L. (2008), Model Selection and Model Averaging, London: Cambridge University Press.
- Constantine, W. and Percival, D. (2016), fractal: Fractal time series modeling and analysis, R package version 2.0-1, <https://CRAN.R-project.org/package=fractal>.

- Cormier, E., Genest, C., and Nešlehová, J. G. (2014), Using B-splines for nonparametric inference on bivariate extreme-value copulas, *Extremes*, 17, 633–659.
- Csörgö, M. and Horváth, L. (1997), Limit Theorems in Change-Point Analysis, Chichester, UK: John Wiley & Sons.
- Darsow, W. W. F., Nguyen, B., and Olsen, E. T. (1992), Copulas and Markov processes, *Illinois Journal of Mathematics*, 36, 600–642.
- Deheuvels, P. (1981), A non parametric test for independence, *Publications de l'Institut de Statistique de l'Université de Paris*, 26, 29–50.
- Dehling, H., Vogel, D., Wendler, M., and Wied, D. (2014), Testing for changes in the rank correlation of time series, tech. rep., ArXiv E-prints, arXiv:1203.4871.
- Devroye, L. (1986), Non-Uniform Random Variate Generation, New York: Springer.
- Durante, F. and Sempi, C. (2010), Copula theory: An introduction, *Copula Theory and Its Applications (Warsaw, 2009)*, ed. by P. Jaworski, F.

Durante, W. K. Härdle, and W. Rychlik, Lecture Notes in Statistics, Berlin: Springer-Verlag, 3–32.

Durante, F. and Sempi, C. (2015), Principles of Copula Theory, Boca Raton, Florida: CRC Press.

Embrechts, P. (2009), Copulas: A personal view, *Journal of Risk and Insurance*, 76, 639–650.

Embrechts, P. and Hofert, M. (2013), A note on generalized inverses, *Mathematical Methods of Operations Research*, 77(3), 423–432.

Embrechts, P., McNeil, A. J., and Straumann, D. (2002), Correlation and dependency in risk management: Properties and pitfalls, *Risk Management: Value at Risk and Beyond*, ed. by M. Dempster, London: Cambridge University Press, 176–223.

Fermanian, J.-D. (2005), Goodness-of-fit tests for copulas, *Journal of Multivariate Analysis*, 95(1), 119–152.

- Fermanian, J.-D. (2013), An overview of the goodness-of-fit test problem for copulas, *Copulae in Mathematical and Quantitative Finance*, ed. by P. Jaworski, F. Durante, and W. K. Härdle, Berlin: Springer, 61–89.
- Fermanian, J.-D. and Wegkamp, M. (2012), Time-dependent copulas, *Journal of Multivariate Analysis*, 110, 19–29.
- Fréchet, M. (1935), Généralisations du théorème des probabilités totales, *Fundamenta Mathematica*, 25, 379–387.
- Fréchet, M. (1951), Sur les tableaux de corrélations dont les marges sont données, *Annales de l'Université de Lyon*, 9, 53–77.
- Frees, E. W. and Valdez, E. A. (1998), Understanding relationships using copulas, *North American Actuarial Journal*, 2, 1–25.
- Genest, C. (1987), Frank's family of bivariate distributions, *Biometrika*, 74(3), 549–555.
- Genest, C. and Favre, A.-C. (2007), Everything you always wanted to know about copula modeling but were afraid to ask, *Journal of Hydrological Engineering*, 12, 347–368.

- Genest, C. and Nešlehová, J. G. (2013), Assessing and Modeling Asymmetry in Bivariate Continuous Data, *Copulae in Mathematical and Quantitative Finance*, ed. by P. Jaworski, F. Durante, and W. K. Härdle, Lecture Notes in Statistics, Berlin: Springer, 91–114.
- Genest, C. and Nešlehová, J. G. (2014), On tests of radial symmetry for bivariate copulas, *Statistical Papers*, 55, 1107–1119.
- Genest, C. and Rémillard, B. (2004), Tests of independence and randomness based on the empirical copula process, *Test*, 13(2), 335–369.
- Genest, C. and Rémillard, B. (2008), Validity of the parametric bootstrap for goodness-of-fit testing in semiparametric models, *Annales de l'Institut Henri Poincaré: Probabilités et Statistiques*, 44, 1096–1127.
- Genest, C., Ghoudi, K., and Rivest, L.-P. (1995), A semiparametric estimation procedure of dependence parameters in multivariate families of distributions, *Biometrika*, 82, 543–552.

- Genest, C., Ghoudi, K., and Rivest, L.-P. (1998), Discussion of “Understanding relationships using copulas”, by E. Frees and E. Valdez, *North American Actuarial Journal*, 3, 143–149.
- Genest, C., Quessy, J.-F., and Rémillard, B. (2006), Goodness-of-fit procedures for copulas models based on the probability integral transformation, *Scandinavian Journal of Statistics*, 33, 337–366.
- Genest, C., Quessy, J.-F., and Rémillard, B. (2007), Asymptotic local efficiency of Cramér-von Mises tests for multivariate independence, *The Annals of Statistics*, 35, 166–191.
- Genest, C., Rémillard, B., and Beaudoin, D. (2009a), Goodness-of-fit tests for copulas: A review and a power study, *Insurance: Mathematics and Economics*, 44, 199–213.
- Genest, C., Gendron, M., and Bourdeau-Brien, M. (2009b), The advent of copulas in finance, *European Journal of Finance*, 15, 609–618.

Genest, C., Kojadinovic, I., Nešlehová, J. G., and Yan, J. (2011a), A goodness-of-fit test for bivariate extreme-value copulas, *Bernoulli*, 17(1), 253–275.

Genest, C., Nešlehová, J. G., and Ruppert, M. (2011b), Comment on the paper by S. Haug, C. Klüppelberg and L. Peng entitled “Statistical models and methods for dependence in insurance data”, *Journal of the Korean Statistical Society*, 40, 141–148.

Genest, C., Nešlehová, J., and Quessy, J.-F. (2012), Tests of symmetry for bivariate copulas, *Annals of the Institute of Statistical Mathematics*, 64, 811–834.

Genest, C., Nešlehová, J. G., and Rémillard, B. (2017), Asymptotic behavior of the empirical multilinear copula process under broad conditions, *Journal of Multivariate Analysis*, 20, 82–110.

Genz, A. et al. (2017), mvtnorm: Multivariate Normal and t Distribution, R package version 1.0-6, <http://CRAN.R-project.org/package=mvtnorm>.

- Ghalanos, A. (2017), rugarch: Univariate GARCH Models, R package version 1.3-8, <https://CRAN.R-project.org/package=rugarch>.
- Gilbert, P. and Varadhan, R. (2016), numDeriv: Accurate Numerical Derivatives, R package version 2016.8-27, <http://CRAN.R-project.org/package=numDeriv>.
- Gombay, E. and Horváth, L. (1999), Change-points and bootstrap, *Environmetrics*, 10(6).
- Grégoire, V., Genest, C., and Gendron, M. (2008), Using copulas to model price dependence in energy markets, *Energy Risk*, 5(5), 58–64.
- Grønneberg, S. and Hjort, N. L. (2014), The copula information criteria, *Scandinavian Journal of Statistics*, 41, 436–459.
- Gumbel, E. J. (1958), Statistics of Extremes, New York: Columbia University Press.
- Harder, M. and Stadtmüller, U. (2017), Testing exchangeability of copulas in arbitrary dimension, *Journal of Nonparametric Statistics*, 29(1), 40–60.

- Higham, N. (2002), Computing the nearest correlation matrix — A problem from finance, *IMA Journal of Numerical Analysis*, 22, 329–343.
- Hofert, M. and Hornik, K. (2017), qrmttools: Tools for Quantitative Risk Management, R package version 0.0-7, [https://CRAN.R-project.org/package=qrmtools](https://CRAN.R-project.org/package=qrmttools).
- Höffding, W. (1940), Massstabvariante Korrelationstheorie, *Schriften des mathematischen Seminars und des Instituts für Angewandte Mathematik der Universität Berlin*, 5, 181–233.
- Holmes, M., Kojadinovic, I., and Quessy, J.-F. (2013), Nonparametric tests for change-point detection à la Gombay and Horváth, *Journal of Multivariate Analysis*, 115, 16–32.
- Ibragimov, R. (2009), Copula based characterizations for higher-order Markov processes, *Econometric Theory*, 25, 819–846.
- Joe, H. (1990), Multivariate concordance, *Journal of Multivariate Analysis*, 35(1), 12–30.

- Joe, H. (1997), Multivariate Models and Dependence Concepts, London: Chapman & Hall.
- Joe, H. (2005), Asymptotic efficiency of the two-stage estimation method for copula-based models, *Journal of Multivariate Analysis*, 94, 401–419.
- Joe, H. (2014), Dependence Modeling with Copulas, Boca Raton, Florida: Chapman & Hall/CRC.
- Jondeau, E. and Rockinger, M. (2006), The copula-garch model of conditional dependencies: An international stock market application, *Journal of International Money and Finance*, 25, 827–853.
- Jordanger, L. A. and Tjøstheim, D. (2014), Model selection of copulas: AIC versus a cross validation copula information criterion, *Statistics & Probability Letters*, 92, 249–255.
- Khoudraji, A. (1995), Contributions à l'étude des copules et à la modélisation des valeurs extrêmes bivariées, PhD thesis, Québec, Canada: Université Laval.

Kojadinovic, I. (2017a), ncp: Some Nonparametric Tests for Change-Point Detection in Possibly Multivariate Observations, R package version 0.1-9, <http://CRAN.R-project.org/package=ncp>.

Kojadinovic, I. (2017b), Some copula inference procedures adapted to the presence of ties, *Computational Statistics and Data Analysis*, 142, 24–41.

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.

Kojadinovic, I. and Yan, J. (2010a), Comparison of three semiparametric methods for estimating dependence parameters in copula models, *Insurance: Mathematics and Economics*, 47, 52–63.

Kojadinovic, I. and Yan, J. (2010b), Modeling multivariate distributions with continuous margins using the copula R package, *Journal of Statistical Software*, 34(9), 1–20.

- Kojadinovic, I. and Yan, J. (2011a), A goodness-of-fit test for multivariate multiparameter copulas based on multiplier central limit theorems, *Statistics and Computing*, 21(1), 17–30.
- Kojadinovic, I. and Yan, J. (2011b), Tests of serial independence for continuous multivariate time series based on a Möbius decomposition of the independence empirical copula process, *Annals of the Institute of Statistical Mathematics*, 63, 347–373.
- Kojadinovic, I. and Yan, J. (2012), A nonparametric test of exchangeability for extreme-value and left-tail decreasing bivariate copulas, *Scandinavian Journal of Statistics*, 39(3), 480–496.
- Kojadinovic, I., Yan, J., and Holmes, M. (2011), Fast large-sample goodness-of-fit for copulas, *Statistica Sinica*, 21(2), 841–871.
- Lehmann, E. L. and Casella, G. (1998), Theory of Point Estimation, New York: Springer.
- Liebscher, E. (2008), Construction of asymmetric multivariate copulas, *Journal of Multivariate Analysis*, 99, 2234–2250.

- Mai, J.-F. and Scherer, M. (2014), Financial Engineering with Copulas Explained, London: Palgrave Macmillan.
- Marshall, A. W. and Olkin, I. (1967), A multivariate exponential distribution, *Journal of the American Statistical Association*, 62, 30–44.
- Mashal, R. and Zeevi, A. (2002), Beyond correlation: Extreme co-movements between financial assets, tech. rep., Columbia School of Business, <https://www0.gsb.columbia.edu/faculty/azeevi/PAPERS/BeyondCorrelation.pdf>.
- McNeil, A. J., Frey, R., and Embrechts, P. (2005), Quantitative Risk Management: Concepts, Techniques and Tools, Princeton, New Jersey: Princeton University Press.
- McNeil, A. J., Frey, R., and Embrechts, P. (2015), Quantitative Risk Management: Concepts, Techniques and Tools, 2nd ed., Princeton, New Jersey: Princeton University Press.

Nason, G. (2016), locits: Tests of stationarity and localized autocovariance, R package version 1.7.3, <https://CRAN.R-project.org/package=locits>.

Nason, G. and Cardinali, A. (2013), costat: Time series costationarity determination, R package version 2.3, <http://CRAN.R-project.org/package=costat>.

Nason, G. P. (2013), A test for second-order stationarity and approximate confidence intervals for localized autocovariances for locally stationary time series, *Journal of the Royal Statistical Society Series B*, 75, 879–904.

Nelsen, R. B. (2006), An Introduction to Copulas, New York: Springer-Verlag.

Pappadà, R., Durante, F., and Salvadori, G. (2017), Quantification of the environmental structural risk with spoiling ties: Is randomization worthwhile? *Stochastic Environmental Research and Risk Assessment*, 31, 2483–2497.

- Patton, A. J. (2006), Modelling asymmetric exchange rate dependence, *International Economic Review*, 47(2), 527–556.
- Patton, A. J. (2012), A review of copula models for economic time series, *Journal of Multivariate Analysis*, 110, 4–18.
- Patton, A. J. (2013), Copula methods for forecasting multivariate time series, *Handbook of Economic Forecasting*, ed. by G. Elliott and A. Timmermann, New York: Springer, 899–960.
- Priestley, M. B. and Subba Rao, T. (1969), A test for stationarity of time series, *Journal of the Royal Statistical Society Series B*, 31, 140–149.
- Quessy, J.-F. (2005), Méthodologie et application des copules: tests d'adéquation, tests d'indépendance, et bornes sur la valeur-à-risque, PhD thesis, Québec, Canada: Université Laval.
- Quessy, J.-F., Saïd, M., and Favre, A.-C. (2013), Multivariate Kendall's tau for change-point detection in copulas, *The Canadian Journal of Statistics*, 41, 65–82.

- Rémillard, B. (2013), Statistical Methods For Financial Engineering, Boca
Baton, Florida: Chapman & Hall/CRC.
- Rémillard, B. (2017), Goodness-of-fit tests for copulas of multivariate time
series, *Econometrics*, 5(13), 1–23.
- Rémillard, B. and Scaillet, O. (2009), Testing for equality between two
copulas, *Journal of Multivariate Analysis*, 100(3), 377–386.
- Rémillard, B., Papageorgiou, N., and Soustra, F. (2012), Copula-based semi-
parametric models for multivariate time series, *Journal of Multivariate
Analysis*, 110, 30–42.
- Rüschenendorf, L. (2009), On the distributional transform, Sklar's Theorem,
and the empirical copula process, *Journal of Statistical Planning and
Inference*, 139(11), 3921–3927.
- Ryan, J. A. and Ulrich, J. M. (2017), xts: eXtensible Time Series, R package
version 0.10-0, <http://CRAN.R-project.org/package=xts>.

- Salvadori, G., De Michele, C., Kottekoda, N. T., and Rosso, R. (2007), Extremes in Nature: An Approach Using Copulas, Water Science and Technology Library, Vol. 56, Berlin: Springer.
- Sarkar, D. (2016), latticeExtra: Extra Graphical Utilities Based on Lattice, R package version 0.6-28, <http://CRAN.R-project.org/package=latticeExtra>.
- Scarsini, M. (1984), On measures of concordance, *Stochastica*, 8(3), 201–218.
- Schmid, F. and Schmidt, R. (2007), Multivariate extensions of Spearman's rho and related statistics, *Statistics & Probability Letters*, 77(4), 407–416.
- Segers, J., Sibuya, M., and Tsukahara, H. (2017), The empirical beta copula, *Journal of Multivariate Analysis*, 155, 35–51.
- Sklar, A. (1996), Random variables, distribution functions, and copulas – A personal look backward and forward, *Distributions with Fixed Marginals and Related Topics*, 28, 1–14.

- Tsukahara, H. (2005), Semiparametric estimation in copula models, *The Canadian Journal of Statistics*, 33(3), 357–375.
- Wild, C. J. and Seber, G. A. F. (1999), *Chance Encounters: A First Course in Data Analysis and Inference*, New York: John Wiley & Sons.
- Yan, J. (2007), Enjoy the Joy of Copulas: With a Package copula, *Journal of Statistical Software*, 21(4), 1–21.