8

Copulas

8.1 Introduction

Copulas are a popular framework for both defining multivariate distributions and modeling multivariate data. A copula characterizes the dependence—and only the dependence—between the components of a multivariate distribution; they can be combined with any set of univariate marginal distributions to form a full joint distribution. Consequently, the use of copulas allows us to take advantage of the wide variety of univariate models that are available.

The primary financial application of copula models is risk assessment and management of portfolios that contain assets which exhibit co-movements in extreme behavior. For example, a pair of assets may have weakly correlated returns, but their largest losses may tend to occur in the same periods. They are commonly applied to portfolios of loans, bonds, and collateralized debt obligations (CDOs). Their misapplication in finance is also well-documented, as referenced in Sect. 8.8.

A *copula* is a multivariate CDF whose univariate marginal distributions are all Uniform(0,1). Suppose that $\boldsymbol{Y} = (Y_1, \ldots, Y_d)$ has a multivariate CDF $F_{\boldsymbol{Y}}$ with continuous marginal univariate CDFs $F_{Y_1}, \ldots, F_{Y_d}$. Then, by Eq. (A.9) in Appendix A.9.2, each of $F_{Y_1}(Y_1), \ldots, F_{Y_d}(Y_d)$ is distributed Uniform(0,1). Therefore, the CDF of $\{F_{Y_1}(Y_1), \ldots, F_{Y_d}(Y_d)\}$ is a copula. This CDF is called the copula of $\boldsymbol{Y}$ and denoted by $C_{\boldsymbol{Y}}$. $C_{\boldsymbol{Y}}$ contains all information about dependencies among the components of $\boldsymbol{Y}$ but has no information about the marginal CDFs of $\boldsymbol{Y}$.

It is easy to find a formula for $C_{\boldsymbol{Y}}$. To avoid technical issues, in this section we will assume that all random variables have continuous, strictly increasing CDFs. More precisely, the CDFs are assumed to be increasing on their support. For example, the standard exponential CDF

D. Ruppert, D.S. Matteson, *Statistics and Data Analysis for Financial Engineering*, Springer Texts in Statistics,
DOI 10.1007/978-1-4939-2614-5_8

$$F(y) = \begin{cases} 1 - e^{-y}, & y \geq 0, \\ 0, & y < 0, \end{cases}$$

has support $[0, \infty)$ and is strictly increasing on that set. The assumption that the CDF is continuous and strictly increasing is reasonable in many financial applications, but it is avoided in more mathematically advanced texts; see Sect. 8.8.

Since C_Y is the CDF of $\{F_{Y_1}(Y_1), \ldots, F_{Y_d}(Y_d)\}$, by the definition of a CDF we have

$$\begin{aligned} C_Y(u_1, \ldots, u_d) &= P\{F_{Y_1}(Y_1) \leq u_1, \ldots, F_{Y_d}(Y_d) \leq u_d\} \\ &= P\left\{Y_1 \leq F_{Y_1}^{-1}(u_1), \ldots, Y_d \leq F_{Y_d}^{-1}(u_d)\right\} \\ &= F_Y\left\{F_{Y_1}^{-1}(u_1), \ldots, F_{Y_d}^{-1}(u_d)\right\}. \end{aligned} \tag{8.1}$$

Next, letting $u_j = F_{Y_j}(y_j)$, for $j = 1, \ldots, d$, in (8.1) we see that

$$F_Y(y_1, \ldots, y_d) = C_Y\{F_{Y_1}(y_1), \ldots, F_{Y_d}(y_d)\}. \tag{8.2}$$

Equation (8.2) is part of a famous theorem due to Sklar which states that the joint CDF F_Y can be decomposed into the copula C_Y, which contains all information about the dependencies among $(Y_1, \ldots, Y_d)$, and the univariate marginal CDFs $F_{Y_1}, \ldots, F_{Y_d}$, which contain all information about the univariate marginal distributions.

Let

$$c_Y(u_1, \ldots, u_d) = \frac{\partial^d}{\partial u_1 \cdots \partial u_d} C_Y(u_1, \ldots, u_d) \tag{8.3}$$

be the density associated with C_Y. By differentiating (8.2), we find that the density of $\boldsymbol{Y}$ is equal to

$$f_Y(y_1, \ldots, y_d) = c_Y\{F_{Y_1}(y_1), \ldots, F_{Y_d}(y_d)\} f_{Y_1}(y_1) \cdots f_{Y_d}(y_d), \tag{8.4}$$

in which $f_{Y_1}, \ldots, f_{Y_d}$ are the univariate marginal densities of $Y_1, \ldots, Y_d$, respectively.

One important property of copulas is that they are invariant to strictly increasing transformations of the component variables. More precisely, suppose that g_j is strictly increasing and $X_j = g_j(Y_j)$ for $j = 1, \ldots, d$. Then $\boldsymbol{X} = (X_1, \ldots, X_d)$ and $\boldsymbol{Y}$ have the same copulas. To see this, first note that the CDF of $\boldsymbol{X}$ is

$$\begin{aligned} F_X(x_1, \ldots, x_d) &= P\{g_1(Y_1) \leq x_1, \ldots, g_d(Y_d) \leq x_d\} \\ &= P\left\{Y_1 \leq g_1^{-1}(x_1), \ldots, Y_d \leq g_d^{-1}(x_d)\right\} \\ &= F_Y\left\{g_1^{-1}(x_1), \ldots, g_d^{-1}(x_d)\right\}, \end{aligned} \tag{8.5}$$

and therefore the CDF of X_j is

$$F_{X_j}(x_j) = F_{Y_j}\left\{g_j^{-1}(x_j)\right\}.$$

Consequently,

$$F_{X_j}^{-1}(u_j) = g_j\left\{F_{Y_j}^{-1}(u_j)\right\}$$

and

$$g_j^{-1}\left\{F_{X_j}^{-1}(u_j)\right\} = F_{Y_j}^{-1}(u_j), \tag{8.6}$$

and by applying (8.1) to $\boldsymbol{X}$, followed by (8.5), (8.6), and then applying (8.1) to $\boldsymbol{Y}$, we conclude that the copula of $\boldsymbol{X}$ is

$$\begin{aligned} C_X(u_1,\ldots,u_d) &= F_X\left\{F_{X_1}^{-1}(u_1),\ldots,F_{X_d}^{-1}(u_d)\right\} \\ &= F_Y\left[g_1^{-1}\left\{F_{X_1}^{-1}(u_1)\right\},\ldots,g_d^{-1}\left\{F_{X_d}^{-1}(u_d)\right\}\right] \\ &= F_Y\left\{F_{Y_1}^{-1}(u_1),\ldots,F_{Y_d}^{-1}(u_d)\right\} \\ &= C_Y(u_1,\ldots,u_d). \end{aligned}$$

8.2 Special Copulas

All d-dimensional copula functions C have domain $[0,1]^d$ and range $[0,1]$. There are three copulas of special interest because they represent independence and two extremes of dependence.

The d-dimensional *independence copula* C_0 is the CDF of d mutually independent Uniform(0,1) random variables. It equals

$$C_0(u_1,\ldots,u_d) = u_1\cdots u_d, \tag{8.7}$$

and the associated density is uniform on $[0,1]^d$; that is, $c_0(u_1,\ldots,u_d) = 1$ on $[0,1]^d$, and zero elsewhere.

The d-dimensional *co-monotonicity copula* C_+ characterizes perfect positive dependence. Let U be Uniform(0,1). Then, the co-monotonicity copula is the CDF of $\boldsymbol{U} = (U,\ldots,U)$; that is, $\boldsymbol{U}$ contains d copies of U so that all of the components of $\boldsymbol{U}$ are equal. Thus,

$$\begin{aligned} C_+(u_1,\ldots,u_d) &= P(U \le u_1,\ldots,U \le u_d) \\ &= P\{U \le \min(u_1,\ldots,u_d)\} = \min(u_1,\ldots,u_d). \end{aligned}$$

The co-monotonicity copula is also an upper bound for all copula functions: $C(u_1,\ldots,u_d) \le C_+(u_1,\ldots,u_d)$, for all $(u_1,\ldots,u_d) \in [0,1]^d$.

The two-dimensional *counter-monotonicity copula* C_- is defined as the CDF of $(U, 1-U)$, which has perfect negative dependence. Therefore,

$$\begin{aligned} C_-(u_1,u_2) &= P(U \le u_1,\ 1-U \le u_2) \\ &= P(1-u_2 \le U \le u_1) = \max(u_1+u_2-1, 0). \end{aligned} \tag{8.8}$$

It is easy to derive the last equality in (8.8). If $1-u_2 > u_1$, then the event $\{1-u_2 \le U \le u_1\}$ is impossible, so the probability is 0. Otherwise, the probability is the length of the interval $(1-u_2, u_1)$, which is u_1+u_2-1. All two-dimensional copula functions are bounded below by (8.8). It is not possible to have a counter-monotonicity copula with $d > 2$. If, for example, U_1 is counter-monotonic to U_2 and U_2 is counter-monotonic to U_3, then U_1 and U_3 will be co-monotonic, not counter-monotonic. However, a lower bound for all copula functions is: $\max(u_1+\cdots+u_d+1-d, 0) \le C(u_1,\ldots,u_d)$, for all $(u_1,\ldots,u_d) \in [0,1]^d$. This lower bound is obtainable only point-wise, but it is not itself a copula function for $d > 2$.

To use copulas to model multivariate dependencies, we next consider parametric families of copulas.

8.3 Gaussian and *t*-Copulas

Multivariate normal and multivariate t-distributions offer a convenient way to generate families of copulas. Let $\boldsymbol{Y} = (Y_1,\ldots,Y_d)$ have a multivariate normal distribution. Since C_Y depends only on the dependencies within $\boldsymbol{Y}$, not the univariate marginal distributions, C_Y depends only on the $d \times d$ correlation matrix of $\boldsymbol{Y}$, which will be denoted by $\boldsymbol{\Omega}$. Therefore, there is a one-to-one correspondence between correlation matrices and Gaussian copulas. The Gaussian copula[1] with correlation matrix $\boldsymbol{\Omega}$ will be denoted $C_{\text{Gauss}}(u_1\ldots,u_d|\boldsymbol{\Omega})$.

If a random vector $\boldsymbol{Y}$ has a Gaussian copula, then $\boldsymbol{Y}$ is said to have a *meta-Gaussian distribution*. This does not, of course, mean that $\boldsymbol{Y}$ has a multivariate Gaussian distribution, since the univariate marginal distributions of $\boldsymbol{Y}$ could be any distributions at all. A d-dimensional Gaussian copula whose correlation matrix is the identity matrix, so that all correlations are zero, is the d-dimensional independence copula. A Gaussian copula will converge to the co-monotonicity copula C_+ if all correlations in $\boldsymbol{\Omega}$ converge to 1. In the bivariate case, as the pair-wise correlation converges to -1, the copula converges to the counter-monotonicity copula C_-.

Similarly, let $C_{\text{t}}(u_1\ldots,u_d|\boldsymbol{\Omega},\nu)$ denote the copula of a random vector that has a multivariate t-distribution with tail index[2] ν and correlation matrix $\boldsymbol{\Omega}$.[3] For multivariate t random vectors the tail index ν affects both the univariate marginal distributions and the tail dependence between components, so ν is a parameter of the t-copula C_{t}. We will see in Sect. 8.6 that ν similarly determines the amount of tail dependence of random vectors that have a t-copula. Such a vector is said to have a *meta-t-distribution*.

---

[1] Gaussian and normal distributions are synonymous and the Gaussian copula may also be referred to as the normal copula, especially in `R` functions.

[2] The tail index parameter for the t-distribution is also commonly referred to as the degrees-of-freedom parameter by its association with the theory of linear regression, and some `R` functions use the abbreviations `df` or `nu`.

[3] There is a minor technical issue here if $\nu \le 2$. In this case, the t-distribution does not have covariance and correlation matrices. However, it still has a scale matrix and we will assume that the scale matrix is equal to some correlation matrix $\boldsymbol{\Omega}$.

8.4 Archimedean Copulas

An *Archimedean copula* with a strict generator has the form

$$C(u_1, \ldots, u_d) = \varphi^{-1}\{\varphi(u_1) + \cdots + \varphi(u_d)\}, \tag{8.9}$$

where the generator function φ satisfies the following conditions

1. φ is a continuous, strictly decreasing, and convex function mapping $[0, 1]$ onto $[0, \infty]$,
2. $\varphi(0) = \infty$, and
3. $\varphi(1) = 0$.

A plot of a generator function is shown in Fig. 8.1 to illustrate these properties. It was generated using the `iPsi()` function from `R`'s `copula` package with the following commands.

```
library(copula)
u = seq(0.000001, 1, length=500)
frank = iPsi(copula=archmCopula(family="frank", param=1), u)
plot(u, frank, type="l", lwd=3, ylab=expression(phi(u)))
abline(h=0) ; abline(v=0)
```

It is possible to relax assumption 2, but then the generator is not called strict and construction of the copula is more complex. The generator function φ is not unique; for example, $a\varphi$, in which a is any positive constant, generates the same copula as φ. The independence copula C_0 is an Archimedean copula with generator function $\varphi(u) = -\log(u)$. There are many families of Archimedean copulas, but we will only look at four, the Frank, Clayton, Gumbel, and Joe copulas.

Notice that in (8.9), the value of $C(u_1, \ldots, u_d)$ is unchanged if we permute $u_1, \ldots, u_d$. A distribution with this property is called *exchangeable*. One consequence of exchangeability is that both Kendall's and Spearman's rank correlation introduced later in Sect. 8.5 are the same for all pairs of variables. Archimedean copulas are most useful in the bivariate case or in applications where we expect all pairs to have similar dependencies.

8.4.1 Frank Copula

The Frank copula has generator

$$\varphi_{\text{Fr}}(u|\theta) = -\log\left(\frac{e^{-\theta u} - 1}{e^{-\theta} - 1}\right), \quad -\infty < \theta < \infty.$$

The inverse generator is

$$\varphi_{\text{Fr}}^{-1}(y|\theta) = -\frac{1}{\theta}\log\left\{e^{-y}(e^{-\theta} - 1) + 1\right\}.$$

Therefore, by (8.9), the bivariate Frank copula is

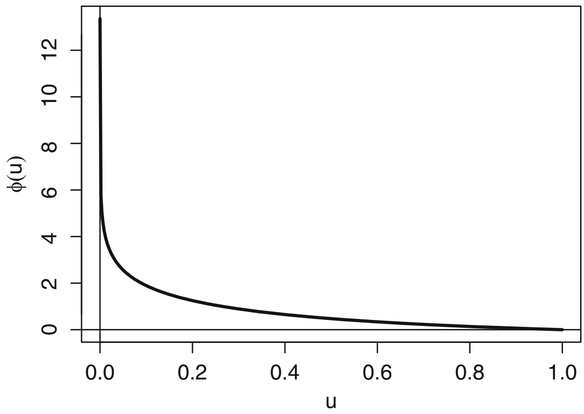

Fig. 8.1. *Generator function for the Frank copula with $\theta = 1$.*

$$C_{\mathrm{Fr}}(u_1, u_2|\theta) = -\frac{1}{\theta}\log\left\{1 + \frac{(e^{-\theta u_1} - 1)(e^{-\theta u_2} - 1)}{e^{-\theta} - 1}\right\}. \tag{8.10}$$

The case $\theta = 0$ requires some care, since plugging this value into (8.10) gives $0/0$. Instead, one must evaluate the limit of (8.10) as $\theta \to 0$. Using the approximations $e^x - 1 \approx x$ and $\log(1+x) \approx x$ as $x \to 0$, one can show that as $\theta \to 0$, $C_{\mathrm{Fr}}(u_1, u_2|\theta) \to u_1 u_2$, the bivariate independence copula C_0. Therefore, for $\theta = 0$ we define the Frank copula to be the independence copula.

It is interesting to study the limits of $C_{\mathrm{Fr}}(u_1, u_2|\theta)$ as $\theta \to \pm\infty$. As $\theta \to -\infty$, the bivariate Frank copula converges to the counter-monotonicity copula C_-. To see this, first note that as $\theta \to -\infty$,

$$C_{\mathrm{Fr}}(u_1, u_2|\theta) \sim -\frac{1}{\theta}\log\left\{1 + e^{-\theta(u_1+u_2-1)}\right\}. \tag{8.11}$$

If $u_1 + u_2 - 1 > 0$, then as $\theta \to -\infty$, the exponent $-\theta(u_1 + u_2 - 1)$ in (8.11) converges to ∞ and

$$\log\left\{1 + e^{-\theta(u_1+u_2-1)}\right\} \sim -\theta(u_1 + u_2 - 1),$$

so that $C_{\mathrm{Fr}}(u_1, u_2|\theta) \to u_1 + u_2 - 1$. Similarly, if $u_1 + u_2 - 1 < 0$, then $-\theta(u_1 + u_2 - 1) \to -\infty$, and $C_{\mathrm{Fr}}(u_1, u_2|\theta) \to 0$. Putting these results together, we see that $C_{\mathrm{Fr}}(u_1, u_2|\theta)$ converges to $\max(0, u_1 + u_2 - 1)$, the counter-monotonicity copula C_-, as $\theta \to -\infty$.

As $\theta \to \infty$, $C_{\mathrm{Fr}}(u_1, u_2|\theta) \to \min(u_1, u_2)$, the co-monotonicity copula C_+. Verification of this is left as an exercise for the reader.

Figure 8.2 contains scatterplots of nine bivariate random samples from various Frank copulas, all with a sample size of 200 and with values of θ that give dependencies ranging from strongly negative to strongly positive. Pseudo-random samples may be generated from the copula distributions discussed in this chapter using the `rCopula()` function from `R`'s `copula` package. The convergence to the counter-monotonicity (co-monotonicity) copula as $\theta \to -\infty$ $(+\infty)$ can be seen in the scatterplots.

```
set.seed(5640)
theta = c(-100, -50, -10, -1, 0, 5, 20, 50, 500)
par(mfrow=c(3,3), cex.axis=1.2, cex.lab=1.2, cex.main=1.2)
for(i in 1:9){
  U = rCopula(n=200,
              copula=archmCopula(family="frank", param=theta[i]))
  plot(U, xlab=expression(u[1]), ylab=expression(u[2]),
       main=eval(substitute(expression(paste(theta," = ",j)),
       list(j = as.character(theta[i])))))
}
```

8.4.2 Clayton Copula

The *Clayton copula*, with generator function $\varphi_{\mathrm{Cl}}(u|\theta) = \frac{1}{\theta}(u^{-\theta} - 1)$, $\theta > 0$, is

$$C_{\mathrm{Cl}}(u_1, \ldots, u_d|\theta) = (u_1^{-\theta} + \cdots + u_d^{-\theta} + 1 - d)^{-1/\theta}.$$

We define the Clayton copula for $\theta = 0$ as

$$\lim_{\theta \downarrow 0} C_{\mathrm{Cl}}(u_1, \ldots, u_d|\theta) = u_1 \cdots u_d$$

which is the independence copula C_0. There is another way to derive this result. As $\theta \downarrow 0$, l'Hôpital's rule shows that the generator $\frac{1}{\theta}(u^{-\theta} - 1)$ converges to $\varphi_{\mathrm{Cl}}(u|\theta \downarrow 0) = -\log(u)$ with inverse $\varphi_{\mathrm{Cl}}^{-1}(y|\theta \downarrow 0) = \exp(-y)$. Therefore,

$$\begin{aligned}\lim_{\theta \downarrow 0} C_{\mathrm{Cl}}(u_1, \ldots, u_d|\theta) &= \varphi_{\mathrm{Cl}}^{-1}\{\varphi_{\mathrm{Cl}}(u_1|\theta \downarrow 0) + \cdots + \varphi_{\mathrm{Cl}}(u_d|\theta \downarrow 0)|\theta \downarrow 0\} \\ &= \exp\{-(-\log u_1 - \cdots - \log u_d)\} = u_1 \cdots u_d.\end{aligned}$$

It is possible to extend the range of θ to include $-1 \le \theta < 0$, but then the generator $(u^{-\theta} - 1)/\theta$ is finite at $u = 0$ in violation of assumption 2, of strict generators. Thus, the generator is not strict if $\theta < 0$. As a result, it is necessary to define $C_{\mathrm{Cl}}(u_1, \ldots, u_d|\theta)$ to equal 0 for small values of u_i in this case. To appreciate this, consider the bivariate Clayton copula. If $-1 \le \theta < 0$, then $u_1^{-\theta} + u_2^{-\theta} - 1 < 0$ occurs when u_1 and u_2 are both small. In these cases, $C_{\mathrm{Cl}}(u_1, u_2|\theta)$ is set equal to 0. Therefore, there is no probability in the region $u_1^{-\theta} + u_2^{-\theta} - 1 < 0$. In the limit, as $\theta \to -1$, there is no probability in the region $u_1 + u_2 < 1$.

As $\theta \to -1$, the bivariate Clayton copula converges to the counter-monotonicity copula C_-, and as $\theta \to \infty$, the Clayton copula converges to the co-monotonicity copula C_+.

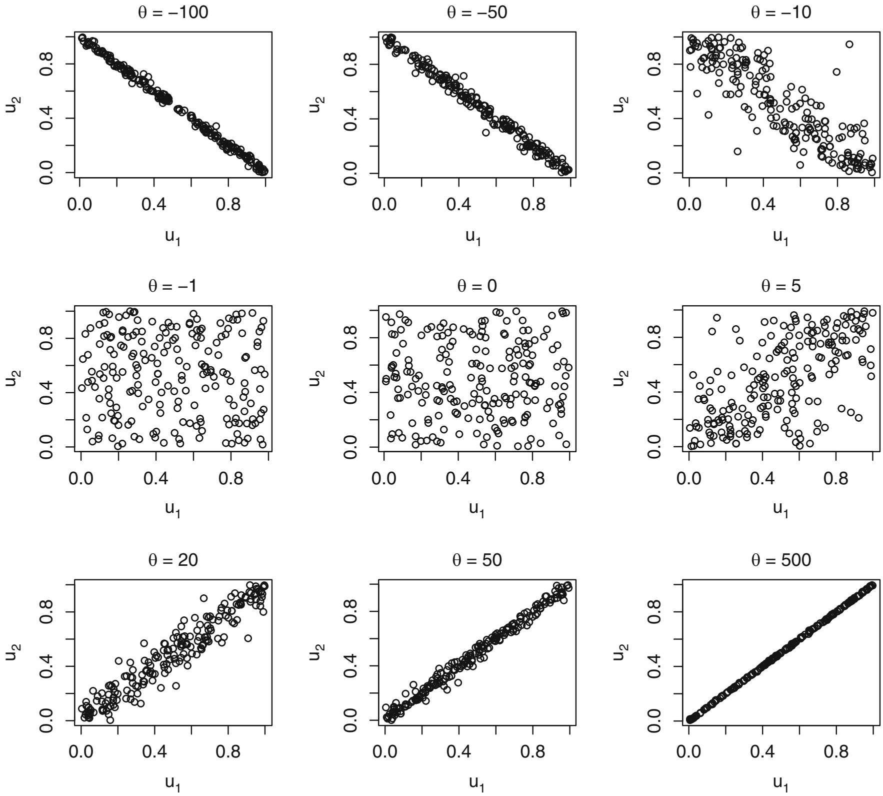

Fig. 8.2. *Bivariate random samples of size 200 from various Frank copulas.*

Figure 8.3 contains scatterplots of nine bivariate random samples from various Clayton copulas, all with a sample size of 200 and with values of θ that give dependencies ranging from counter-monotonicity to co-monotonicity.

```
set.seed(5640)
theta = c(-0.98, -0.7, -0.3, -0.1, 0.1, 1, 5, 15, 100)
par(mfrow=c(3,3), cex.axis=1.2, cex.lab=1.2, cex.main=1.2)
for(i in 1:9){
  U = rCopula(n=200,
              copula=archmCopula(family="clayton", param=theta[i]))
  plot(U, xlab=expression(u[1]), ylab=expression(u[2]),
       main=eval(substitute(expression(paste(theta," = ",j)),
       list(j = as.character(theta[i])))))
}
```

Comparing Figs. 8.2 and 8.3, we see that the Frank and Clayton copulas are rather different when the amount of dependence is somewhere between these two extremes. In particular, the Clayton copula's exclusion of the region $u_1^{-\theta} + u_2^{-\theta} - 1 < 0$ when $\theta < 0$ is evident, especially in the example with

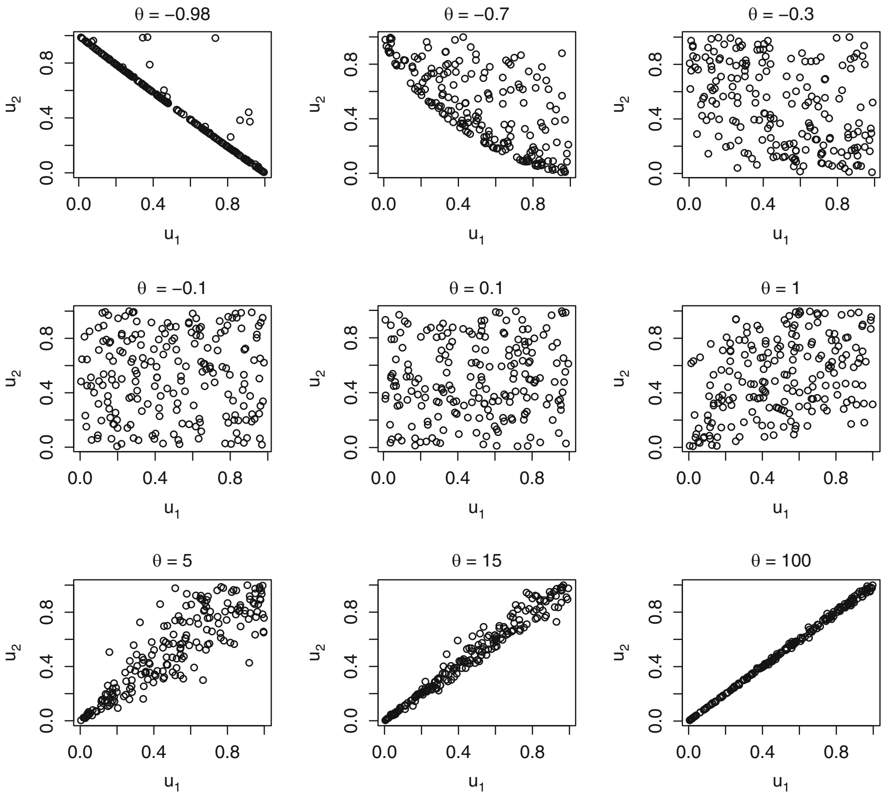

Fig. 8.3. *Bivariate random samples of size 200 from various Clayton copulas.*

$\theta = -0.7$. In contrast, the Frank copula has positive probability on the entire unit square. The Frank copula is symmetric about the diagonal from $(0, 1)$ to $(1, 0)$, but the Clayton copula does not have this symmetry.

8.4.3 Gumbel Copula

The Gumbel copula has the generator $\varphi_{\text{Gu}}(u|\theta) = (-\log u)^\theta$, $\theta \geq 1$, and consequently is equal to

$$C_{\text{Gu}}(u_1, \ldots, u_d|\theta) = \exp\left[-\left\{(-\log u_1)^\theta + \cdots + (-\log u_d)^\theta\right\}^{1/\theta}\right].$$

The Gumbel copula is the independence copula C_0 when $\theta = 1$, and converges to the co-monotonicity copula C_+ as $\theta \to \infty$, but the Gumbel copula cannot have negative dependence.

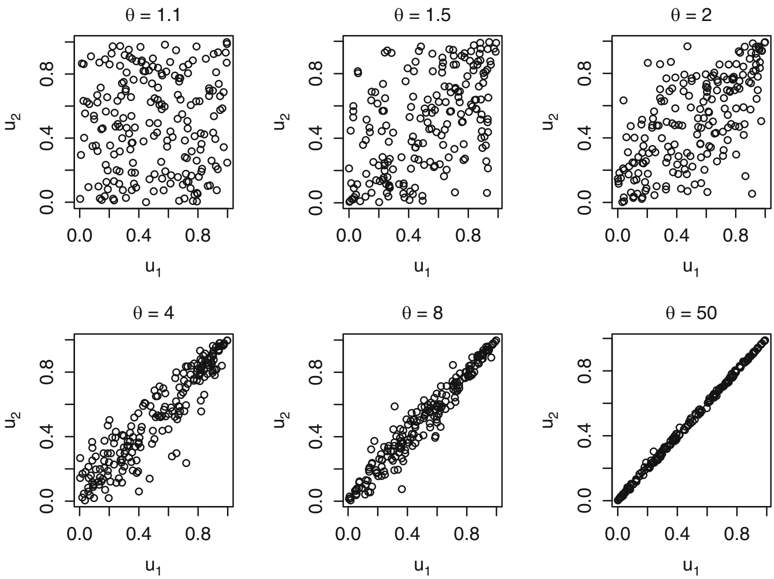

Fig. 8.4. *Bivariate random samples of size 200 from various Gumbel copulas.*

Figure 8.4 contains scatterplots of six bivariate random samples from various Gumbel copulas, with a sample size of 200 and with values of θ that give dependencies ranging from near independence to strong positive dependence.

```
set.seed(5640)
theta = c(1.1, 1.5, 2, 4, 8, 50)
par(mfrow=c(2,3), cex.axis=1.2, cex.lab=1.2, cex.main=1.2)
for(i in 1:6){
  U = rCopula(n=200,
                copula=archmCopula(family="gumbel", param=theta[i]))
  plot(U, xlab=expression(u[1]), ylab=expression(u[2]),
        main=eval(substitute(expression(paste(theta," = ",j)),
        list(j = as.character(theta[i])))))
}
```

8.4.4 Joe Copula

The Joe copula is similar to the Gumbel copula; it cannot have negative dependence, but it allows even stronger upper tail dependence and is closer to being the reverse of the Clayton copula in the positive dependence case. The Joe copula has the generator $\varphi_{\text{Joe}}(u|\theta) = -\log\{1-(1-u)^\theta\}$, $\theta \geq 1$. In the bivariate case, the Joe copula is equal to

$$C_{\text{Joe}}(u_1, u_2|\theta) = 1 - \left[(1-u_1)^\theta + (1-u_2)^\theta - (1-u_1)^\theta(1-u_2)^\theta\right]^{1/\theta}.$$

The Joe copula is the independence copula C_0 when $\theta = 1$, and converges to the co-monotonicity copula C_+ as $\theta \to \infty$.

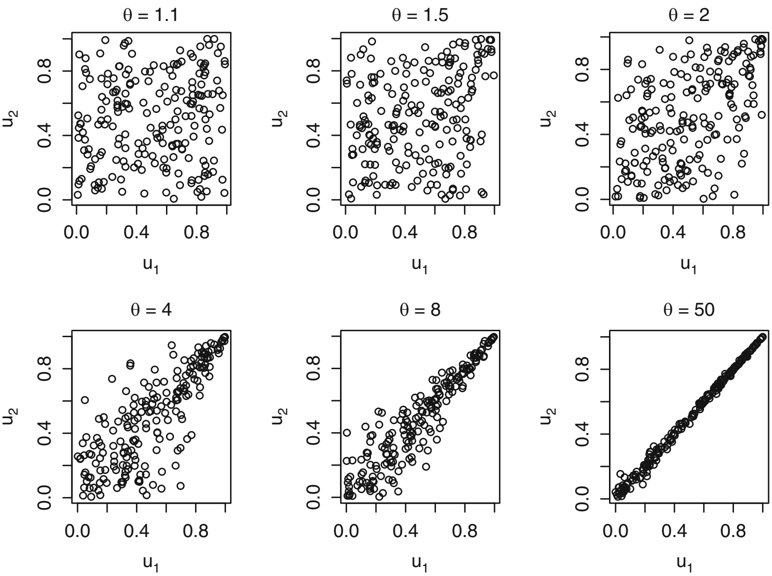

Fig. 8.5. *Bivariate random samples of size 200 from various Joe copulas.*

Figure 8.5 contains scatterplots of six bivariate random samples from various Joe copulas, with a sample size of 200 and with values of θ that give dependencies ranging from near independence to strong positive dependence.

```
set.seed(5640)
theta = c(1.1, 1.5, 2, 4, 8, 50)
par(mfrow=c(2,3), cex.axis=1.2, cex.lab=1.2, cex.main=1.2)
for(i in 1:6){
  U = rCopula(n=200,
              copula=archmCopula(family="joe", param=theta[i]))
  plot(U, xlab=expression(u[1]), ylab=expression(u[2]),
       main=eval(substitute(expression(paste(theta," = ",j)),
       list(j = as.character(theta[i])))))
}
```

In applications, it is useful that the different copula families have different properties, since this increases our ability to find a copula that fits the data adequately.

8.5 Rank Correlation

The Pearson correlation coefficient defined by (4.4) is not convenient for fitting copulas to data, since it depends on the univariate marginal distributions as well as the copula. Rank correlation coefficients remedy this problem, since they depend only on the copula.

For each variable, the ranks of that variable are determined by ordering the observations from smallest to largest and giving the smallest rank 1, the next-smallest rank 2, and so forth. In other words, if $Y_1, \ldots, Y_n$ is a sample, then the *rank* of Y_i in the sample is equal to 1 if Y_i is the smallest observation, 2 if Y_i is the second smallest, and so forth. More mathematically, the rank of Y_i can also be defined by the formula

$$\text{rank}(Y_i) = \sum_{j=1}^{n} I(Y_j \le Y_i), \tag{8.12}$$

which counts the number of observations (including Y_i itself) that are less than or equal to Y_i. A *rank statistic* is a statistic that depends on the data only through the ranks. A key property of ranks is that they are unchanged by strictly monotonic transformations of the variables. In particular, the ranks are unchanged by transforming each variable by its CDF, so the distribution of any rank statistic depends only on the copula of the observations, not on the univariate marginal distributions.

We will be concerned with rank statistics that measure statistical association between pairs of variables. These statistics are called *rank correlations.* There are two rank correlation coefficients in widespread usage, Kendall's tau and Spearman's rho.

8.5.1 Kendall's Tau

Let (Y_1, Y_2) be a bivariate random vector and let (Y_1^*, Y_2^*) be an independent copy of (Y_1, Y_2). Then (Y_1, Y_2) and (Y_1^*, Y_2^*) are called a *concordant pair* if the ranking of Y_1 relative to Y_1^* is the same as the ranking of Y_2 relative to Y_2^*, that is, either $Y_1 > Y_1^*$ and $Y_2 > Y_2^*$ or $Y_1 < Y_1^*$ and $Y_2 < Y_2^*$. In either case, $(Y_1 - Y_1^*)(Y_2 - Y_2^*) > 0$. Similarly, (Y_1, Y_2) and (Y_1^*, Y_2^*) are called a *discordant pair* if $(Y_1 - Y_1^*)(Y_2 - Y_2^*) < 0$. *Kendall's tau* is the probability of a concordant pair minus the probability of a discordant pair. Therefore, Kendall's tau for (Y_1, Y_2) is

$$\begin{aligned} \rho_\tau(Y_1, Y_2) &= P\{(Y_1 - Y_1^*)(Y_2 - Y_2^*) > 0\} - P\{(Y_1 - Y_1^*)(Y_2 - Y_2^*) < 0\} \\ &= E\left[\text{sign}\{(Y_1 - Y_1^*)(Y_2 - Y_2^*)\}\right], \end{aligned} \tag{8.13}$$

where the *sign function* is

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

It is clear from (8.13) that ρ_τ is symmetric in its arguments and takes values in $[-1, 1]$. It is easy to check that if g and h are increasing functions, then

$$\rho_\tau\{g(Y_1), h(Y_2)\} = \rho_\tau(Y_1, Y_2). \tag{8.14}$$

Stated differently, Kendall's tau is invariant to monotonically increasing transformations. If g and h are the marginal CDFs of Y_1 and Y_2, then the left-hand side of (8.14) is the value of Kendall's tau for a pair of random variables distributed according to the copula of (Y_1, Y_2). This shows that Kendall's tau depends only on the copula of a bivariate random vector. For a random vector $\boldsymbol{Y}$, we define the *Kendall's tau correlation matrix* $\boldsymbol{\Omega}_\tau$ to be the matrix whose (j,k) entry is Kendall's tau for the jth and kth components of $\boldsymbol{Y}$, that is $[\boldsymbol{\Omega}_\tau(\boldsymbol{Y})]_{jk} = \rho_\tau(Y_j, Y_k)$.

If we have a bivariate sample $\boldsymbol{Y}_{1:n} = \{(Y_{i,1}, Y_{i,2}) : i = 1, \ldots, n\}$, then the sample Kendall's tau is

$$\widehat{\rho}_\tau(\boldsymbol{Y}_{1:n}) = \binom{n}{2}^{-1} \sum_{1 \le i < j \le n} \operatorname{sign}\{(Y_{i,1} - Y_{j,1})(Y_{i,2} - Y_{j,2})\}. \tag{8.15}$$

Note that $\binom{n}{2}$ is the number of summands in (8.15), so $\widehat{\rho}_\tau$ is the average of $\operatorname{sign}\{(Y_{i,1} - Y_{j,1})\ (Y_{i,2} - Y_{j,2})\}$ across all distinct pairs of observations and is a sample version of (8.13). The sample Kendall's tau correlation matrix is defined analogously to $\boldsymbol{\Omega}_\tau$.

8.5.2 Spearman's Rank Correlation Coefficient

For a sample, Spearman's correlation coefficient is simply the usual Pearson correlation calculated from the marginal ranks of the data. For a distribution (that is, an infinite population rather than a finite sample), both variables are transformed by their univariate marginal CDFs and then the Pearson correlation is computed for the transformed variables. Transforming a random variable by its CDF is analogous to computing the ranks of a variable in a finite sample.

Stated differently, Spearman's rank correlation coefficient, also called *Spearman's rho*, for a bivariate random vector (Y_1, Y_2) will be denoted as $\rho_S(Y_1, Y_2)$ and is defined to be the Pearson correlation coefficient of $\{F_{Y_1}(Y_1), F_{Y_2}(Y_2)\}$:

$$\rho_S(Y_1, Y_2) = \operatorname{Corr}\{F_{Y_1}(Y_1), F_{Y_2}(Y_2)\}.$$

Since the joint CDF of $\{F_{Y_1}(Y_1), F_{Y_2}(Y_2)\}$ is the copula of (Y_1, Y_2), Spearman's rho, like Kendall's tau, depends only on the copula function.

The sample version of Spearman's correlation coefficient can be computed from the ranks of the data and for a bivariate sample $\boldsymbol{Y}_{1:n} = \{(Y_{i,1}, Y_{i,2}) : i = 1, \ldots, n\}$, is

$$\widehat{\rho}_S(\boldsymbol{Y}_{1:n}) = \frac{12}{n(n^2-1)} \sum_{i=1}^{n} \left\{\operatorname{rank}(Y_{i,1}) - \frac{n+1}{2}\right\}\left\{\operatorname{rank}(Y_{i,2}) - \frac{n+1}{2}\right\}. \tag{8.16}$$

The set of ranks for any variable is, of course, the integers 1 to n, and hence $(n+1)/2$ is the mean of its ranks. It can be shown that $\widehat{\rho}_S(\boldsymbol{Y}_{1:n})$ is the sample Pearson correlation between the ranks of $\{Y_{i,1}\}$ and the ranks of $\{Y_{i,2}\}$.[4]

If $\boldsymbol{Y} = (Y_1, \ldots, Y_d)$ is a random vector, then the *Spearman's correlation matrix* $\boldsymbol{\Omega}_S$ of $\boldsymbol{Y}$ is the correlation matrix of $\{F_{Y_1}(Y_1), \ldots, F_{Y_d}(Y_d)\}$ and contains the Spearman's correlation coefficients for all pairs of components of $\boldsymbol{Y}$, such that $[\boldsymbol{\Omega}_S(\boldsymbol{Y})]_{jk} = \rho_S(Y_j, Y_k)$, for all $j, k = 1, \ldots, d$. The sample Spearman's correlation matrix is defined analogously.

8.6 Tail Dependence

Tail dependence measures association between the extreme values of two random variables and depends only on their copula. We will start with lower tail dependence, which uses extremes in the lower tail. Suppose that $\boldsymbol{Y} = (Y_1, Y_2)$ is a bivariate random vector with copula C_Y. Then the *coefficient of lower tail dependence* is denoted by λ_ℓ and defined as

$$\lambda_\ell := \lim_{q \downarrow 0} P\big\{Y_2 \le F_{Y_2}^{-1}(q) \,|\, Y_1 \le F_{Y_1}^{-1}(q)\big\} \tag{8.17}$$

$$= \lim_{q \downarrow 0} \frac{P\big\{Y_1 \le F_{Y_1}^{-1}(q), Y_2 \le F_{Y_2}^{-1}(q)\big\}}{P\big\{Y_1 \le F_{Y_1}^{-1}(q)\big\}} \tag{8.18}$$

$$= \lim_{q \downarrow 0} \frac{P\{F_{Y_1}(Y_1) \le q, F_{Y_2}(Y_2) \le q\}}{P\{F_{Y_1}(Y_1) \le q\}} \tag{8.19}$$

$$= \lim_{q \downarrow 0} \frac{C_Y(q, q)}{q}. \tag{8.20}$$

It is helpful to look at these equations individually. As elsewhere in this chapter, for simplicity we are assuming that F_{Y_1} and F_{Y_2} are strictly increasing on their supports and therefore have inverses.

First, (8.17) defines λ_ℓ as the limit as $q \downarrow 0$ of the conditional probability that Y_2 is less than or equal to its qth quantile, given that Y_1 is less than or equal to its qth quantile. Since we are taking a limit as $q \downarrow 0$, we are looking at the extreme left tail. What happens if Y_1 and Y_2 are independent? Then $P(Y_2 \le y_2 \,|\, Y_1 \le y_1) = P(Y_2 \le y_2)$ for all y_1 and y_2. Therefore, the conditional probability in (8.17) equals the unconditional probability $P(Y_2 \le F_{Y_2}^{-1}(q))$ and this probability converges to 0 as $q \downarrow 0$. Therefore, $\lambda_\ell = 0$ implies that in the extreme left tail, Y_1 and Y_2 behave as if they were independent.

Equation (8.18) is just the definition of conditional probability. Equation (8.19) is simply (8.18) after applying the probability transformation to each variable. The numerator in (8.19) is the copula by definition, and the

[4] If there are ties, then ranks are averaged among tied observations. For example, if there are two observations tied for smallest, then they each get a rank of 1.5. When there are ties, these results must be modified.

denominator in (8.20) is the result of $F_{Y_1}(Y_1)$ being distributed Uniform(0,1); see (A.9).

Deriving formulas for λ_ℓ for Gaussian and t-copulas is a topic best left for more advanced books. Here we give only the results; see Sect. 8.8 for further reading. For any bivariate Gaussian copula C_{Gauss} with $\rho \neq 1$, $\lambda_\ell = 0$, that is, Gaussian copulas do not have tail dependence except in the extreme case of perfect positive correlation. For a bivariate t-copula C_{t} with tail index ν and correlation ρ,

$$\lambda_\ell = 2F_{t,\nu+1}\left\{-\sqrt{\frac{(\nu+1)(1-\rho)}{1+\rho}}\right\}, \tag{8.21}$$

where $F_{t,\nu+1}$ is the CDF of the t-distribution with tail index $\nu + 1$.

Since $F_{t,\nu+1}(-\infty) = 0$, we see that $\lambda_\ell \to 0$ as $\nu \to \infty$, which makes sense since the t-copula converges to a Gaussian copula as $\nu \to \infty$. Also, $\lambda_\ell \to 0$ as $\rho \to -1$, which is also not too surprising, since $\rho = -1$ is perfect *negative* dependence and λ_ℓ measures *positive* tail dependence.

The *coefficient of upper tail dependence* λ_u is

$$\lambda_u := \lim_{q \uparrow 1} P\{Y_2 \geq F_{Y_2}^{-1}(q) \,|\, Y_1 \geq F_{Y_1}^{-1}(q)\} \tag{8.22}$$

$$= 2 - \lim_{q \uparrow 1} \frac{1 - C_Y(q,q)}{1-q}. \tag{8.23}$$

We see that λ_u is defined analogously to λ_ℓ; λ_u is the limit as $q \uparrow 1$ of the conditional probability that Y_2 is greater than or equal to its qth quantile, given that Y_1 is greater than or equal to its qth quantile. Deriving (8.23) is left as an exercise for the interested reader.

For Gaussian and t-copula, $\lambda_u = \lambda_\ell$, so that $\lambda_u = 0$ for any Gaussian copula and for a t-copula, λ_ℓ is given by the right-hand side of (8.21). Coefficients of tail dependence for t-copulas are plotted in Fig. 8.6. One can see $\lambda_\ell = \lambda_u$ depends strongly on both ρ and ν. For the independence copula C_0, λ_ℓ and λ_u are both equal to 0, and for the co-monotonicity copula C_+, both are equal to 1.

```
rho = seq(-1,1, by=0.01)
df = c(1, 4, 25, 240)
x1 = -sqrt((df[1]+1)*(1-rho)/(1+rho))
lambda1 = 2*pt(x1,df[1]+1)
x4 = -sqrt((df[2]+1)*(1-rho)/(1+rho))
lambda4 = 2*pt(x4,df[2]+1)
x25 = -sqrt((df[3]+1)*(1-rho)/(1+rho))
lambda25 = 2*pt(x25,df[3]+1)
x250 = -sqrt((df[4]+1)*(1-rho)/(1+rho))
lambda250 = 2*pt(x250,df[4]+1)
par(mfrow=c(1,1), lwd=2, cex.axis=1.2, cex.lab=1.2)
plot(rho, lambda1, type="l", lty=1, xlab=expression(rho),
```

```
    ylab=expression(lambda[l]==lambda[u]))
lines(rho, lambda4, lty=2)
lines(rho, lambda25, lty=3)
lines(rho, lambda250, lty=4)
legend("topleft", c(expression(nu==1), expression(nu==4),
      expression(nu==25), expression(nu==250)), lty=1:4)
```

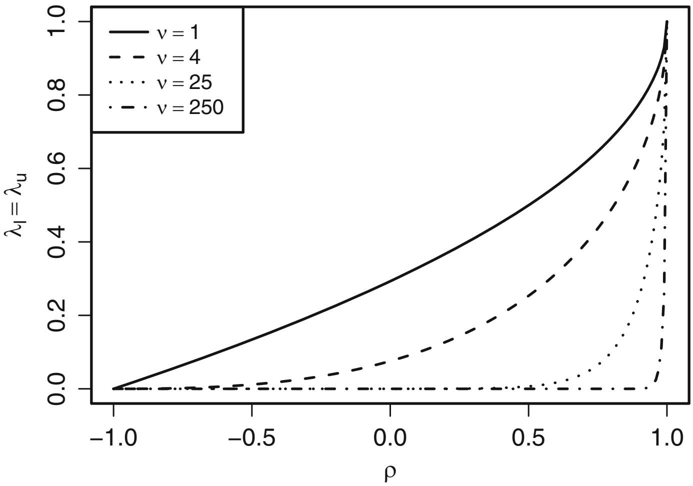

Fig. 8.6. *Coefficients of tail dependence for bivariate t-copulas as functions of* ρ *for* $\nu = 1, 4, 25,$ *and* 250.

Knowing whether or not there is tail dependence is important for risk management. If there are no tail dependencies among the returns on the assets in a portfolio, then there is little risk of simultaneous very negative returns, and the risk of an extreme negative return on the portfolio is low. Conversely, if there are tail dependencies, then the likelihood of extreme negative returns occurring simultaneously on several assets in the portfolio can be high. As such, tail dependencies should be considered when assessing the diversification and risk of any portfolio.

8.7 Calibrating Copulas

Assume that we have an i.i.d. sample $\boldsymbol{Y}_{1:n} = \{(Y_{i,1}, \ldots, Y_{i,d}) : i = 1, \ldots, n\}$, and we wish to estimate the copula of $\boldsymbol{Y}$ and perhaps its univariate marginal distributions as well.

An important task is choosing a copula model. The various copula models differ notably from each other. For example, some have tail dependence and others do not. The Gumbel copula and Joe copula allow only positive dependence or independence. The Clayton copula with negative dependence excludes the region where both u_1 and u_2 are small. As will be seen in this section, an appropriate copula model can be selected via AIC, and by using graphical techniques.

8.7.1 Maximum Likelihood

Suppose we have parametric models $F_{Y_1}(\cdot \mid \boldsymbol{\theta}_1), \ldots, F_{Y_d}(\cdot \mid \boldsymbol{\theta}_d)$ for the marginal CDFs as well as a parametric model $c_Y(\cdot|\boldsymbol{\theta}_C)$ for the copula density. By taking logs of (8.4), we find that the log-likelihood is

$$\log\{L(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d, \boldsymbol{\theta}_C)\} = \sum_{i=1}^{n} \Big(\log[c_Y\{F_{Y_1}(Y_{i,1}|\boldsymbol{\theta}_1), \ldots, F_{Y_d}(Y_{i,d}|\boldsymbol{\theta}_d)|\boldsymbol{\theta}_C\}] + \log\{f_{Y_1}(Y_{i,1}|\boldsymbol{\theta}_1)\} + \cdots + \log\{f_{Y_d}(Y_{i,d}|\boldsymbol{\theta}_d)\}\Big). \quad (8.24)$$

Maximum likelihood estimation finds the maximum of $\log\{L(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d, \boldsymbol{\theta}_C)\}$ over the entire set of parameters $(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d, \boldsymbol{\theta}_C)$.

There are two potential problems with maximum likelihood estimation. First, because of the large number of parameters, especially for large values of d, numerically maximizing $\log\{L(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d, \boldsymbol{\theta}_C)\}$ can be challenging. This difficulty can be ameliorated by the use of starting values that are close to the MLEs. The pseudo-maximum likelihood estimates discussed in the next section are easier to compute than the MLE and can be used either as an alternative to the MLE or as starting values for the MLE.

Second, maximum likelihood estimation requires parametric models for both the copula and the univariate marginal distributions. If any of the univariate marginal distributions are not well fit by a convenient parametric family, this may cause biases in the estimated parameters of both the univariate marginal distributions and the copula. The semiparametric approach to pseudo-maximum likelihood estimation, where the univariate marginal distributions are estimated nonparametrically, provides a remedy to this problem.

8.7.2 Pseudo-Maximum Likelihood

Pseudo-maximum likelihood estimation is a two-step procedure. In the first step, each of the d univariate marginal distribution functions is estimated, one at a time. Let $\widehat{F}_{Y_j}$ be the estimate of the jth univariate marginal CDF, $j = 1, \ldots, d$. In the second step,

$$\sum_{i=1}^{n} \log\Big[c_Y\Big\{\widehat{F}_{Y_1}(Y_{i,1}), \ldots, \widehat{F}_{Y_d}(Y_{i,d})|\boldsymbol{\theta}_C\Big\}\Big] \quad (8.25)$$

is maximized over $\boldsymbol{\theta}_C$. Note that (8.25) is obtained from (8.24) by deleting terms that do not depend on $\boldsymbol{\theta}_C$ and replacing the univariate marginal CDFs by estimates. By estimating parameters in the univariate marginal distributions and in the copula separately, the pseudo-maximum likelihood approach avoids a high-dimensional optimization.

There are two approaches to the first step, parametric and nonparametric. In the parametric approach, parametric models $F_{Y_1}(\cdot\,|\,\boldsymbol{\theta}_1), \ldots, F_{Y_d}(\cdot\,|\,\boldsymbol{\theta}_d)$ for the univariate marginal CDFs are assumed as in maximum likelihood estimation. The data $Y_{1,j}, \ldots, Y_{n,j}$ for the jth variate are used to estimate $\boldsymbol{\theta}_j$, usually by maximum likelihood as discussed in Chap. 5. Then, $\widehat{F}_{Y_j}(\cdot) = F_{Y_j}(\cdot|\widehat{\boldsymbol{\theta}}_j)$. In the nonparametric approach, $\widehat{F}_{Y_j}$ is estimated by the empirical CDF of $Y_{1,j}, \ldots, Y_{n,j}$, except that the divisor n in (4.2) is replaced by $n+1$ so that

$$\widehat{F}_{Y_j}(y) = \frac{\sum_{i=1}^{n} I\{Y_{i,j} \le y\}}{n+1}. \tag{8.26}$$

With this modified divisor, the maximum value of $\widehat{F}_{Y_j}(Y_{i,j})$ is $n/(n+1)$ rather than 1. Avoiding a value of 1 is essential when, as is often the case, $c_Y(u_1, \ldots, u_d|\boldsymbol{\theta}_C) = \infty$ if some of $u_1, \ldots, u_d$ are equal to 1.

When both steps are parametric, the estimation method is called *parametric pseudo-maximum likelihood*. The combination of a nonparametric first step and a parametric second step is called *semiparametric pseudo-maximum likelihood*.

In the second step of pseudo-maximum likelihood, the maximization can be difficult when $\boldsymbol{\theta}_C$ is high-dimensional. For example, if one uses a Gaussian or t-copula, then there are $d(d-1)/2$ correlation parameters. One way to solve this problem is to assume some structure among the correlations. An extreme case of this is the *equi-correlation model* where all non-diagonal elements of the correlation matrix have a common value, call it ρ. If one is reluctant to assume some type of structured correlation matrix, then it is essential to have good starting values for the correlation matrix when maximizing (8.25). For Gaussian and t-copulas, starting values can be obtained via rank correlations as discussed in the next section.

The values $\widehat{F}_{Y_j}(Y_{i,j})$, $i = 1, \ldots, n$ and $j = 1, \ldots, d$, will be called the *uniform-transformed variables*, since they should be distributed approximately Uniform(0,1). The multivariate empirical CDF [see Eq. (A.38)] of the uniform-transformed variables is called the *empirical copula* and is a nonparametric estimate of the copula function. The empirical copula is useful for checking the goodness of fit of parametric copula models; see Example 8.1.

8.7.3 Calibrating Meta-Gaussian and Meta-t-Distributions

Gaussian Copulas

Rank correlation can be useful for estimating the parameters of a copula. Suppose $\boldsymbol{Y}_{1:n} = \{(Y_{i,1}, \ldots, Y_{i,d}) : i = 1, \ldots, n\}$, is an i.i.d. sample from a meta-Gaussian distribution. Then its copula is $C_{\text{Gauss}}(\,\cdot\,|\boldsymbol{\Omega})$ for some correlation matrix $\boldsymbol{\Omega}$. To estimate the distribution of $\boldsymbol{Y}$, we need to estimate the univariate marginal distributions and $\boldsymbol{\Omega}$. The marginal distribution can be estimated by the methods discussed in Chap. 5. Result (8.28) in the following theorem shows that $\boldsymbol{\Omega}$ can be estimated by the sample Spearman's correlation matrix.

Result 8.1 *Let* $\boldsymbol{Y} = (Y_1, \ldots, Y_d)$ *have a meta-Gaussian distribution with continuous univariate marginal distributions and copula* $C_{\text{Gauss}}(\,\cdot\,|\boldsymbol{\Omega})$*, and let* $\Omega_{ij} = [\boldsymbol{\Omega}]_{ij}$*. Then*

$$\rho_\tau(Y_i, Y_j) = \frac{2}{\pi}\arcsin(\Omega_{ij}), \text{ and} \tag{8.27}$$

$$\rho_S(Y_i, Y_j) = \frac{6}{\pi}\arcsin(\Omega_{ij}/2) \approx \Omega_{ij}. \tag{8.28}$$

Suppose, instead, that $\boldsymbol{Y}$ *has a meta-t-distribution with continuous univariate marginal distributions and copula* $C_{\text{t}}(\,\cdot\,|\boldsymbol{\Omega}, \nu)$*. Then (8.27) still holds, but (8.28) does not hold.*

The approximation in (8.28) uses the result that

$$\frac{6}{\pi}\arcsin(x/2) \approx x \text{ for } |x| \leq 1. \tag{8.29}$$

The left- and right-hand sides of (8.29) are equal when $x = -1, 0, 1$, and their maximum difference over the range $-1 \leq x \leq 1$ is 0.018. However, the relative error $\left\{\frac{6}{\pi}\arcsin(x/2) - x\right\} / \frac{6}{\pi}\arcsin(x/2)$ can be larger, as much as 0.047, and is largest near $x = 0$.

By (8.28), the sample Spearman's rank correlation matrix $\widehat{\boldsymbol{\Omega}}(\boldsymbol{Y}_{1:n})$ can be used as an estimate of the correlation matrix $\boldsymbol{\Omega}$ associated with $C_{\text{Gauss}}(\,\cdot\,|\boldsymbol{\Omega})$. This estimate could be the final one or could be used as a starting value for numeric maximum likelihood or pseudo-maximum likelihood estimation.

t-Copulas

If $\boldsymbol{Y}_{1:n} = \{(Y_{i,1}, \ldots, Y_{i,d}) : i = 1, \ldots, n\}$ is a sample from a distribution with a t-copula $C_{\text{t}}(\,\cdot\,|\boldsymbol{\Omega}, \nu)$ then we can use (8.27) and the sample Kendall's tau correlations to estimate $\boldsymbol{\Omega}$. Let $\widehat{\Omega}_{\tau,jk}$ be the sample Kendall's tau correlation of $\{Y_{1,j}, \ldots, Y_{n,j}\}$ and $\{Y_{1,k}, \ldots, Y_{n,k}\}$, the jth and kth components, and let $\widetilde{\boldsymbol{\Omega}}^{**}$ be defined such that $[\widetilde{\boldsymbol{\Omega}}^{**}]_{jk} = \sin\{\frac{\pi}{2}\widehat{\Omega}_{\tau,jk}\}$ Then $\widetilde{\boldsymbol{\Omega}}^{**}$ will have two of the three properties of a correlation matrix; it will be symmetric, with all diagonal entries equal to 1. However, it may not be positive definite, or even semidefinite, because some of its eigenvalues may be negative.

If all of the eigenvalues of $\widetilde{\boldsymbol{\Omega}}^{**}$ are positive, then we will use $\widetilde{\boldsymbol{\Omega}}^{**}$ to estimate $\boldsymbol{\Omega}$. Otherwise, we alter $\widetilde{\boldsymbol{\Omega}}^{**}$ slightly to make it positive definite. By (A.50),

$$\widetilde{\boldsymbol{\Omega}}^{**} = \boldsymbol{O}\,\mathrm{diag}(\lambda_i)\,\boldsymbol{O}^{\mathsf{T}},$$

where $\boldsymbol{O}$ is an orthogonal matrix whose columns are the eigenvectors of $\widetilde{\boldsymbol{\Omega}}^{**}$ and $\lambda_1, \ldots, \lambda_d$ are the corresponding eigenvalues. We then define

$$\widetilde{\boldsymbol{\Omega}}^{*} = \boldsymbol{O}\,\mathrm{diag}\{\max(\epsilon, \lambda_i)\}\,\boldsymbol{O}^{\mathsf{T}},$$

where ϵ is some small positive quantity, for example, $\epsilon = 0.001$. Now, $\widetilde{\boldsymbol{\Omega}}^{*}$ is symmetric and positive definite, but its diagonal elements, $\widetilde{\Omega}^{*}_{ii}$, $i = 1, \ldots, d$, may not be equal to 1. This problem is easily fixed; multiply the ith row and the ith column of $\widetilde{\boldsymbol{\Omega}}^{*}$ by $\big(\widetilde{\Omega}^{*}_{ii}\big)^{-1/2}$, for $i = 1, \ldots, d$. The final result, which we denote as $\widetilde{\boldsymbol{\Omega}}$, is a bona fide correlation matrix; that is, it is symmetric, positive definite, and it has all diagonal entries equal to 1.

After $\boldsymbol{\Omega}$ has been estimated by $\widetilde{\boldsymbol{\Omega}}$, an estimate of the tail index ν is still needed. One can be obtained by plugging $\widetilde{\boldsymbol{\Omega}}$ into the log-likelihood (8.25) and then maximizing over ν.

Example 8.1. Flows in pipelines

In this example, we will continue the analysis of the pipeline flows data introduced in Example 4.2. Only the flows in the first two pipelines will be used.

In a fully parametric pseudo-likelihood analysis, the univariate skewed t-model will be used for flows 1 and 2. Let $\widehat{U}_{1,j}, \ldots, \widehat{U}_{n,j}$ be the flows in pipeline j, $j = 1, 2$, transformed by their estimated skewed-t CDFs. We will call the $\widehat{U}_{i,j}$ "uniform-transformed flows." Define $\widehat{Z}_{i,j} = \Phi^{-1}(\widehat{U}_{i,j})$, where Φ^{-1} is the standard normal quantile function. The $\widehat{Z}_{i,j}$ should each be approximately $N(0,1)$-distributed, and we will call them "normal-transformed flows."

```
library(copula)
library(sn)
dat = read.csv("FlowData.csv")
dat = dat/10000
n = nrow(dat)
x1 = dat$Flow1
fit1 = st.mple(matrix(1,n,1), y=x1, dp=c(mean(x1), sd(x1), 0, 10))
est1 = fit1$dp
u1 = pst(x1, dp=est1)
x2 = dat$Flow2
fit2 = st.mple(matrix(1,n,1), y=x2, dp=c(mean(x2), sd(x2), 0, 10))
est2 = fit2$dp
u2 = pst(x2, dp=est2)
U.hat = cbind(u1, u2)
z1 = qnorm(u1)
z2 = qnorm(u2)
Z.hat = cbind(z1, z2)
```

Both sets of uniform-transformed flows should be approximately Uniform(0,1). Figure 8.7 shows density histograms of both samples of uniform-transformed flows as well as their scatterplot and two-dimensional KDE density contours. The histograms show some deviations from uniform distributions, which suggests that the skewed-t model may not provide adequate fits and that a semiparametric pseudo-maximum likelihood approach might be tried—this is considered below. However, the deviations may be due to random variation.

```
library(ks)
fhatU = kde(x=U.hat, H=Hscv(x=U.hat))
par(mfrow=c(2,2), cex.axis=1.2, cex.lab=1.2, cex.main=1.2)
hist(u1, main="(a)", xlab=expression(hat(U)[1]), freq = FALSE)
hist(u2, main="(b)", xlab=expression(hat(U)[2]), freq = FALSE)
plot(u1, u2, main="(c)", xlab = expression(hat(U)[1]),
     ylab = expression(hat(U)[2]), mgp = c(2.5, 1, 0))
plot(fhatU, drawpoints=FALSE, drawlabels=FALSE,
     cont=seq(10, 80, 10), main="(d)", xlab=expression(hat(U)[1]),
     ylab=expression(hat(U)[2]), mgp = c(2.5, 1, 0))
```

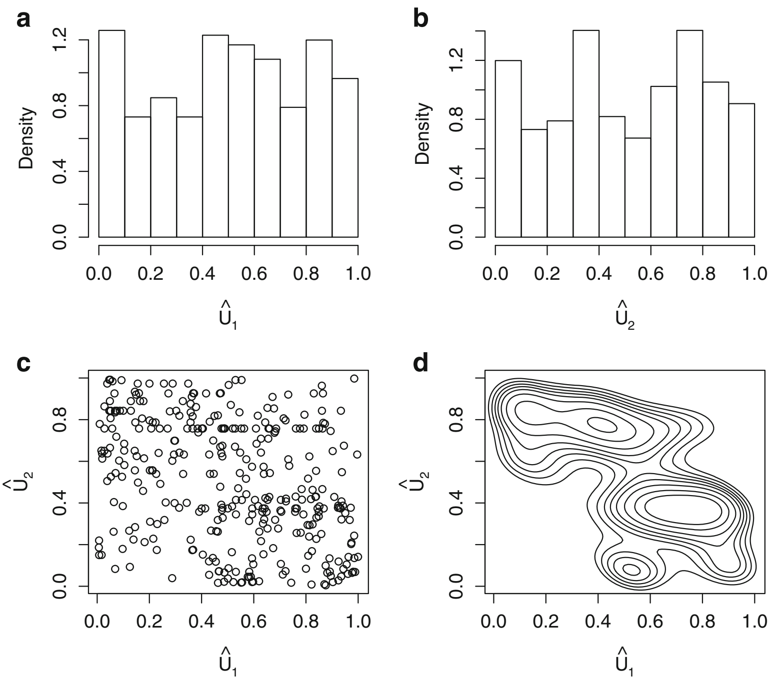

Fig. 8.7. *Pipeline data. Density histograms (**a**) and (**b**) and a scatterplot (**c**) of the uniform-transformed flows. The empirical copula $\widehat{C}$ is the empirical CDF of the data in (**c**). Contours (**d**) from an estimated copula density $\widehat{c}$ via a two-dimensional KDE of (**c**).*

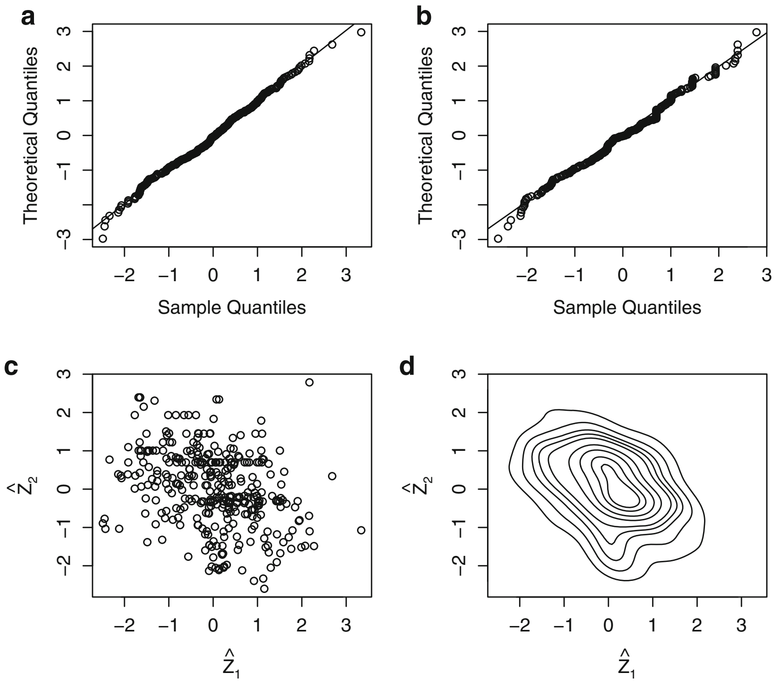

Fig. 8.8. *Pipeline data. Normal quantile plots (**a**) and (**b**), a scatterplot (**c**) and KDE density contours for the normal-transformed flows.*

The scatterplot in Fig. 8.7 shows some negative association as the data are somewhat concentrated along the diagonal from top left to bottom right. Thus, the Gumbel copula and Joe copula, which cannot have negative dependence, are not appropriate. Also, the Clayton copula may not fit well either, since the scatterplot shows data in the region where both $\widehat{U}_1$ and $\widehat{U}_2$ have small values, but this region is excluded by a Clayton copula with negative dependence. We will soon see that AIC agrees with these conclusions from a graphical analysis, since the Clayton model has higher (worse) AIC values compared to the Gaussian, t, and Frank copula models.

Figure 8.8 shows that the normal-transformed flows have approximately linear normal quantile plots, which would be expected if the estimated univariate marginal CDFs were adequate fits. Their scatterplot and KDE density contours again show negative association.

```
fhatZ = kde(x=Z.hat, H=Hscv(x=Z.hat))
par(mfrow=c(2,2), cex.axis=1.2, cex.lab=1.2, cex.main=1.2)
qqnorm(z1, datax=T, main="(a)") ; qqline(z1)
qqnorm(z2, datax=T, main="(b)") ; qqline(z2)
plot(z1, z2, main="(c)", xlab = expression(hat(Z)[1]),
     ylab = expression(hat(Z)[2]), mgp = c(2.5, 1, 0))
plot(fhatZ, drawpoints=FALSE, drawlabels=FALSE,
```

```
     cont=seq(10, 90, 10), main="(d)", xlab=expression(hat(Z)[1]),
     ylab=expression(hat(Z)[2]), mgp = c(2.5, 1, 0))
```

We will assume for now that the two flows have a meta-Gaussian distribution. There are three ways to estimate the correlation in their Gaussian copula. The first, Spearman's rank correlation, is estimated -0.357. The second, which uses (8.27) is $\sin(\widehat{\rho}_\tau \pi/2)$, where $\widehat{\rho}_\tau$ is the sample Kendall's tau rank correlation; its value is -0.371. The third, Pearson correlation of the normal-transformed flows, is -0.335. There is reasonably close agreement among the three values, especially relative to their uncertainties; for example, the approximate 95 % confidence interval for the Pearson correlation of the normal-transformed flows is $(-0.426, -0.238)$, and the other two estimate are well within this interval.

```
cor.test(u1, u2, method="spearman")
cor.test(u1, u2, method="kendall")
sin(-0.242*pi/2)
cor.test(u1, u2, method="pearson")
cor.test(z1, z2, method="pearson")
```

```
 Pearson's product-moment correlation

 data:  z1 and z2
 t = -6.56, df = 340, p-value = 2.003e-10
 alternative hypothesis: true correlation is not equal to 0
 95 percent confidence interval:
  -0.426 -0.238
 sample estimates:
    cor
 -0.335
```

Four parametric copulas were fit to the uniform-transformed flows: t, Gaussian, Frank and Clayton. Estimation of the copula distributions discussed in this chapter may be performed using the `fitCopula()` function from `R`'s `copula` package. The Gumbel and Joe copulas are not considered since they only allow positive dependence and these data show negative dependence; attempting to fit these models results in numerical failures. Since we used parametric estimates to transform the flows, we are fitting the copulas by parametric pseudo-maximum likelihood.

```
omega = -0.371
Ct = fitCopula(copula=tCopula(dim = 2), data=U.hat,
               method="ml", start=c(omega, 10))
Ct@estimate
loglikCopula(param=Ct@estimate, x=U.hat, copula=tCopula(dim = 2))
-2*.Last.value + 2*length(Ct@estimate)
#
Cgauss = fitCopula(copula=normalCopula(dim = 2), data=U.hat,
                   method="ml", start=c(omega))
Cgauss@estimate
```

```
loglikCopula(param=Cgauss@estimate, x=U.hat,
             copula=normalCopula(dim = 2))
-2*.Last.value + 2*length(Cgauss@estimate)
#
Cfr = fitCopula(copula=frankCopula(1, dim=2), data=U.hat,
                method="ml")
Cfr@estimate
loglikCopula(param=Cfr@estimate, x=U.hat,
             copula=frankCopula(dim = 2))
-2*.Last.value + 2*length(Cfr@estimate)
#
Ccl = fitCopula(copula=claytonCopula(1, dim=2), data=U.hat,
                method="ml")
Ccl@estimate
loglikCopula(param=Ccl@estimate, x=U.hat,
             copula=claytonCopula(dim = 2))
-2*.Last.value + 2*length(Ccl@estimate)
```

The results are summarized in Table 8.1. Looking at the maximized log-likelihood values, we see that the Frank copula fits best since it minimizes AIC, but the t and Gaussian fit reasonably well. Figure 8.9 shows the uniform-transformed flows scatterplot and contours of the distribution functions of five copulas: the independence copula and the four estimated parametric copulas; the empirical copula contours have been overlaid for comparison. The t-copula is similar to the Gaussian since $\widehat{\nu} = 22.247$ is large. The Frank copula fits best in the sense that its contours are closest to those of the empirical copula. This is in agreement with the AIC values.

Table 8.1. *Estimates of copula parameters, maximized log-likelihood, and AIC using the uniform-transformed pipeline flow data.*

Copula family	Estimates	Maximized log-likelihood	AIC
t	$\widehat{\rho} = -0.340$	20.98	−37.96
	$\widehat{\nu} = 22.247$		
Gaussian	$\widehat{\rho} = -0.331$	20.36	−38.71
Frank	$\widehat{\theta} = -2.249$	23.07	−44.13
Clayton	$\widehat{\theta} = -0.166$	9.86	−17.72

```
par(mfrow=c(2,3), mgp = c(2.5, 1, 0))
plot(u1, u2, main="Uniform-Transformed Data",
     xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]))
Udex = (1:n)/(n+1)
Cn = C.n(u = cbind(rep(Udex, n), rep(Udex, each=n)) , U = U.hat,
         offset=0, method="C")
EmpCop = expression(contour(Udex,Udex,matrix(Cn,n,n), col=2, add=T))
#
```

```
contour(normalCopula(param=0,dim=2), pCopula, main=expression(C[0]),
        xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]))
eval(EmpCop)
#
contour(tCopula(param=Ct@estimate[1], dim=2,
                df=round(Ct@estimate[2])),
        pCopula, main = expression(hat(C)[t]),
        xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]))
eval(EmpCop)
#
contour(normalCopula(param=Cgauss@estimate[1], dim = 2),
        pCopula, main = expression(hat(C)[Gauss]),
        xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]))
eval(EmpCop)
#
contour(frankCopula(param=Cfr@estimate[1], dim = 2),
        pCopula, main = expression(hat(C)[Fr]),
        xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]))
eval(EmpCop)
#
contour(claytonCopula(param=Ccl@estimate[1], dim = 2),
        pCopula, main = expression(hat(C)[Cl]),
        xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]))
eval(EmpCop)
```

The analysis in the previous paragraph was repeated with the flows transformed by their empirical CDFs. Doing this yielded the semiparametric pseudo-maximum likelihood estimates. Since the results were very similar to those for parametric pseudo-maximum likelihood estimates, they are not presented here. □

8.8 Bibliographic Notes

For discussion of Archimedean copula with non-strict generators, see McNeil, Frey, and Embrechts (2005). These authors discuss a number of other topics in more detail than is done here. They discuss methods defining nonexchangeable Archimedean copulas. The coefficients of tail dependence for Gaussian and t-copulas are derived in their Sect. 5.2. The theorem and calibration methods in Sect. 8.7.3 are discussed in their Sect. 5.5.

Cherubini et al. (2004) treat the application of copulas to finance. Joe (1997) and Nelsen (2007) are standard references on copulas. Chapter 4 of Mari and Kotz (2001) discusses additional copula families.

Li (2000) developed a well-known but controversial model for credit risk using exponentially distributed default times with a Gaussian copula. An article in *Wired* magazine states that Li's Gaussian copula model was "a quick—and

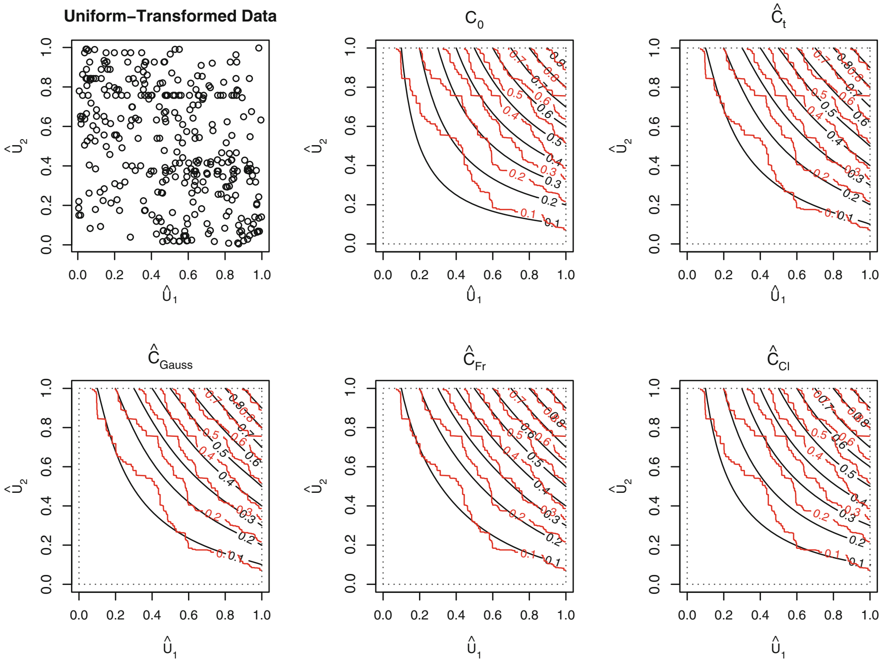

Fig. 8.9. *Uniform-transformed flows for pipeline data. Scatterplot; independence copula contours and four fitted copula contours via parametric models, versus the empirical copula contours.*

fatally flawed—way to assess risk" (Salmon 2009); in particular, the model does not include tail dependence. Duffie and Singleton's (2003, Section 10.4) also discusses copula-based methods for modeling dependent default times.

8.9 R Lab

8.9.1 Simulating from Copula Models

Run the R code that appears below to generate data from a copula. Line 1 loads the `copula` package. Lines 2–3 defines a copula object. At this point, nothing is done with the copula object—it is simply defined. However, the copula object is used in line 5 to generate a random sample from the specified copula model. The remaining lines create a scatterplot matrix of the sample and print its sample Pearson correlation matrix.

```
1 library(copula)
2 cop_t_dim3 = tCopula(dim = 3, param = c(-0.6,0.75,0),
3                      dispstr = "un", df = 1)
4 set.seed(5640)
```

```
5 rand_t_cop = rCopula(n = 500, copula = cop_t_dim3)
6 pairs(rand_t_cop)
7 cor(rand_t_cop)
```

You can use R's help to learn more about the functions `tCopula()` and `rCopula()`.

Problem 1 *Consider the* R *code above.*

(a) What type of copula model has been sampled? Give the copula family, the correlation matrix, and any other parameters that specify the copula.
(b) What is the sample size?

Problem 2 *Examine the scatterplot matrix (generated by line 6) and answer the questions below. Include the scatterplot matrix with your answer.*

(a) Components 2 and 3 are uncorrelated. Do they appear independent? Why or why not?
(b) Do you see signs of tail dependence? If so, where?
(c) What are the effects of dependence upon the plots?
(d) The nonzero correlations in the copula do not have the same values as the corresponding sample correlations. Do you think this is just due to random variation or is something else going on? If there is another cause besides random variation, what might that be? To help answer this question, you can get confidence intervals for the Pearson correlation: For example,

```
8 cor.test(rand_t_cop[,1],rand_t_cop[,3])
```

will give a confidence interval (95 percent by default) for the correlation (Pearson by default) between components 1 and 3. Does this confidence interval include 0.75*?*

Lines 9–10 in the R code below defines a normal (Gaussian) copula. Lines 11–13 define a multivariate distribution by specifying its copula and its marginal distributions—the copula is the one just defined. Line 15 generates a random sample of size 1,000 from this distribution, which has three components. The remaining lines create a scatterplot matrix and kernel estimates of the marginal densities for each component.

```
9  cop_normal_dim3 = normalCopula(dim = 3, param = c(-0.6,0.75,0),
10                                  dispstr = "un")
11 mvdc_normal = mvdc(copula = cop_normal_dim3, margins = rep("exp",3),
12                     paramMargins = list(list(rate=2), list(rate=3),
13                                          list(rate=4)))
14 set.seed(5640)
15 rand_mvdc = rMvdc(n = 1000, mvdc = mvdc_normal)
16 pairs(rand_mvdc)
17 par(mfrow = c(2,2))
18 for(i in 1:3) plot(density(rand_mvdc[,i]))
```

Problem 3 *Run the* R *code above to generate a random sample.*

(a) What are the marginal distributions of the three components in `rand_mvdc`*? What are their expected values?*
(b) Are the second and third components independent? Why or why not?

8.9.2 Fitting Copula Models to Bivariate Return Data

In this section, you will fit copula models to a bivariate data set of daily returns on IBM stock and the S&P 500 index.

First, you will fit a model with univariate marginal t-distributions and a t-copula. The model has three degrees-of-freedom (tail index) parameters, one for each of the two univariate models and a third for the copula. This means that the univariate distributions can have different tail indices and that their tail indices are independent of the tail dependence from the copula.

Run the following R code to load the data and necessary libraries, fit univariate t-distributions to the two components, and convert estimated scale parameters to estimated standard deviations:

```
1  library(MASS)     #  for fitdistr() and kde2d() functions
2  library(copula)   #  for copula functions
3  library(fGarch)   #  for standardized t density
4  netRtns = read.csv("IBM_SP500_04_14_daily_netRtns.csv", header = T)
5  ibm = netRtns[,2]
6  sp500 = netRtns[,3]
7  est.ibm = as.numeric( fitdistr(ibm,"t")$estimate )
8  est.sp500 = as.numeric( fitdistr(sp500,"t")$estimate )
9  est.ibm[2] = est.ibm[2] * sqrt( est.ibm[3] / (est.ibm[3]-2) )
10 est.sp500[2] = est.sp500[2] * sqrt(est.sp500[3] / (est.sp500[3]-2) )
```

The univariate estimates will be used as starting values when the meta-t-distribution is fit by maximum likelihood. You also need an estimate of the correlation coefficient in the t-copula. This can be obtained using Kendall's tau. Run the following code and complete line 12 so that `omega` is the estimate of the Pearson correlation based on Kendall's tau.

```
11 cor_tau = cor(ibm, sp500, method = "kendall")
12 omega =
```

Problem 4 *How did you complete line 12 of the code? What was the computed value of* `omega`*?*

Next, define the t-copula using `omega` as the correlation parameter and 4 as the degrees-of-freedom (tail index) parameter.

```
cop_t_dim2 = tCopula(omega, dim = 2, dispstr = "un", df = 4)
```

Now fit copulas to the uniform-transformed data.

```
data1 = cbind(pstd(ibm, est.ibm[1], est.ibm[2], est.ibm[3]),
              pstd(sp500, est.sp500[1], est.sp500[2], est.sp500[3]))
n = nrow(netRtns) ; n
data2 = cbind(rank(ibm)/(n+1), rank(sp500)/(n+1))
ft1 = fitCopula(cop_t_dim2, data1, method="ml", start=c(omega,4) )
ft2 = fitCopula(cop_t_dim2, data2, method="ml", start=c(omega,4) )
```

Problem 5

(a) *Explain the difference between methods used to obtain the two estimates* `ft1` *and* `ft2`*.*
(b) *Do the two estimates seem significantly different (in a practical sense)?*

The next step defines a meta-t-distribution by specifying its t-copula and its univariate marginal distributions. Values for the parameters in the univariate margins are also specified. The values of the copula parameter were already defined in the previous step.

```
mvdc_t_t = mvdc( cop_t_dim2, c("std","std"), list(
           list(mean=est.ibm[1],sd=est.ibm[2],nu=est.ibm[3]),
           list(mean=est.sp500[1],sd=est.sp500[2],nu=est.sp500[3])))
```

Now fit the meta t-distribution. Be patient. This takes awhile; for instance, it took one minute on my laptop. The elapsed time in minutes will be printed.

```
start = c(est.ibm, est.sp500, ft1@estimate)
objFn = function(param) -loglikMvdc(param,cbind(ibm,sp500),mvdc_t_t)
tic = proc.time()
ft = optim(start, objFn, method="L-BFGS-B",
           lower = c(-.1,0.001,2.2, -0.1,0.001,2.2,  0.2,2.5),
           upper = c( .1,   10, 15,  0.1,   10, 15,  0.9, 15) )
toc = proc.time()
total_time = toc - tic ; total_time[3]/60
```

Lower and upper bounds are used to constrain the algorithm to stay inside a region where the log-likelihood is defined and finite. The function `fitMvdc()` in the `copula` package does not allow setting lower and upper bounds and did not converge on this problem.

Problem 6

(a) *What are the estimates of the copula parameters in* `fit_cop`*?*
(b) *What are the estimates of the parameters in the univariate marginal distributions?*

(c) *Was the estimation method maximum likelihood, semiparametric pseudo-maximum likelihood, or parametric pseudo-maximum likelihood?*
(d) *Estimate the coefficient of lower tail dependence for this copula.*

Now fit normal (Gaussian), Frank, Clayton, Gumbel and Joe copulas to the data.

```
fnorm = fitCopula(copula=normalCopula(dim=2),data=data1,method="ml")
ffrank = fitCopula(copula = frankCopula(3, dim = 2),
                     data = data1, method = "ml")
fclayton = fitCopula(copula = claytonCopula(1, dim=2),
                     data = data1, method = "ml")
fgumbel = fitCopula(copula = gumbelCopula(3, dim=2),
                    data = data1, method = "ml")
fjoe = fitCopula(copula=joeCopula(2,dim=2),data=data1,method="ml")
```

The estimated copulas (CDFs) will be compared with the empirical copula.

```
Udex = (1:n)/(n+1)
Cn = C.n(u=cbind(rep(Udex,n),rep(Udex,each=n)), U=data1, method="C")
EmpCop = expression(contour(Udex, Udex, matrix(Cn, n, n),
                            col = 2, add = TRUE))
par(mfrow=c(2,3),  mgp = c(2.5,1,0))
contour(tCopula(param=ft$par[7],dim=2,df=round(ft$par[8])),
        pCopula, main = expression(hat(C)[t]),
        xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]) )
eval(EmpCop)
contour(normalCopula(param=fnorm@estimate[1], dim = 2),
        pCopula, main = expression(hat(C)[Gauss]),
        xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]) )
eval(EmpCop)
contour(frankCopula(param=ffrank@estimate[1], dim = 2),
        pCopula, main = expression(hat(C)[Fr]),
        xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]) )
eval(EmpCop)
contour(claytonCopula(param=fclayton@estimate[1], dim = 2),
        pCopula, main = expression(hat(C)[Cl]),
        xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]) )
eval(EmpCop)
contour(gumbelCopula(param=fgumbel@estimate[1], dim = 2),
        pCopula, main = expression(hat(C)[Gu]),
        xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]) )
eval(EmpCop)
contour(joeCopula(param=fjoe@estimate[1], dim = 2),
        pCopula, main = expression(hat(C)[Joe]),
        xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]) )
eval(EmpCop)
```

Problem 7 *Do you see any difference between the parametric estimates of the copula? If so, which seem closest to the empirical copula? Include the plot with your work.*

A two-dimensional KDE of the copula's density will be compared with the parametric density estimates (PDFs).

```
par(mfrow=c(2,3),  mgp = c(2.5,1,0))
contour(tCopula(param=ft$par[7],dim=2,df=round(ft$par[8])),
        dCopula, main = expression(hat(c)[t]),
  nlevels=25, xlab=expression(hat(U)[1]),ylab=expression(hat(U)[2]))
contour(kde2d(data1[,1],data1[,2]), col = 2, add = TRUE)
contour(normalCopula(param=fnorm@estimate[1], dim = 2),
        dCopula, main = expression(hat(c)[Gauss]),
  nlevels=25, xlab=expression(hat(U)[1]),ylab=expression(hat(U)[2]))
contour(kde2d(data1[,1],data1[,2]), col = 2, add = TRUE)
contour(frankCopula(param=ffrank@estimate[1], dim = 2),
        dCopula, main = expression(hat(c)[Fr]),
  nlevels=25, xlab=expression(hat(U)[1]),ylab=expression(hat(U)[2]))
contour(kde2d(data1[,1],data1[,2]), col = 2, add = TRUE)
contour(claytonCopula(param=fclayton@estimate[1], dim = 2),
        dCopula, main = expression(hat(c)[Cl]),
  nlevels=25, xlab=expression(hat(U)[1]),ylab=expression(hat(U)[2]))
contour(kde2d(data1[,1],data1[,2]), col = 2, add = TRUE)
contour(gumbelCopula(param=fgumbel@estimate[1], dim = 2),
        dCopula, main = expression(hat(c)[Gu]),
  nlevels=25, xlab=expression(hat(U)[1]),ylab=expression(hat(U)[2]))
contour(kde2d(data1[,1],data1[,2]), col = 2, add = TRUE)
contour(joeCopula(param=fjoe@estimate[1], dim = 2),
        dCopula, main = expression(hat(c)[Joe]),
  nlevels=25, xlab=expression(hat(U)[1]),ylab=expression(hat(U)[2]))
contour(kde2d(data1[,1],data1[,2]), col = 2, add = TRUE)
```

Problem 8 *Do you see any difference between the parametric estimates of the copula density? If so, which seem closest to the KDE? Include the plot with your work.*

Problem 9 *Find AIC for the t, (Gaussian), Frank, Clayton, Gumbel and Joe copulas. Which copula model fits best by AIC? (Hint: The* `fitCopula()` *function returns the log-likelihood.)*

8.10 Exercises

1. Kendall's tau rank correlation between X and Y is 0.55. Both X and Y are positive. What is Kendall's tau between X and $1/Y$? What is Kendall's tau between $1/X$ and $1/Y$?

2. Suppose that X is Uniform(0,1) and $Y = X^2$. Then the Spearman rank correlation and the Kendall's tau between X and Y will both equal 1, but the Pearson correlation between X and Y will be less than 1. Explain why.
3. Show that an Archimedean copula with generator function $\varphi(u) = -\log(u)$ is equal to the independence copula C_0. Does the same hold when the natural logarithm is replaced by the common logarithm, i.e., $\varphi(u) = -\log_{10}(u)$?
4. The co-monotonicity copula C_+ is not an Archimedean copula; however, in the two-dimensional case, the counter-monotonicity copula $C_-(u_1, u_2) = \max(u_1 + u_2 - 1, 0)$ is. What is its generator function?
5. Show that the generator of a Frank copula

$$\varphi_{\mathrm{Fr}}(u|\theta) = -\log\left\{\frac{e^{-\theta u} - 1}{e^{-\theta} - 1}\right\}, \quad \theta \in \{(-\infty, 0) \cup (0, \infty)\},$$

satisfies assumptions 1–3 of a strict generator.
6. Show that as $\theta \to \infty$, $C_{\mathrm{Fr}}(u_1, u_2|\theta) \to \min(u_1, u_2)$, the co-monotonicity copula C_+.
7. Suppose that $\varphi_1, \ldots, \varphi_k$ are k strict generator functions and define a new generator φ as a convex combination of these k generators, that is

$$\varphi(u) = a_1\varphi_1(u) + \cdots + a_k\varphi_k(u),$$

in which $a_1, \ldots, a_k$ are any non-negative constants which sum to 1. Show that $\varphi(u)$ is a strict generator function. For the case in which $k = 2$, what is the corresponding copula function for $\varphi(u)$?
8. Let $\varphi(u|\theta) = (1 - u)^\theta$, for some $\theta \geq 1$, and show that for the two-dimensional case this generates the copula

$$C(u_1, u_2|\theta) = \max[0, 1 - \{(1 - u_1)^\theta + (1 - u_2)^\theta\}^{1/\theta}].$$

Further, show that as $\theta \to \infty$, $C(u_1, u_2|\theta) \to \min(u_1, u_2)$, the co-monotonicity copula C_+, and that as $\theta \to 1$, $C(u_1, u_2|\theta) \to \max(u_1 + u_2 - 1, 0)$, the counter-monotonicity copula C_-.
9. A convex combination of k joint CDFs is itself a joint CDF (finite mixture), but is a convex combination of k copula functions a copula function itself?
10. Suppose $\mathbf{Y} = (Y_1, \ldots, Y_d)$ has a meta-Gaussian distribution with continuous marginal distributions and copula $C^{Gauss}(\cdot|\mathbf{\Omega})$. Show that if $\rho_\tau(Y_i, Y_j) = 0$ then Y_i and Y_j are independent.

References

Cherubini, U., Luciano, E., and Vecchiato, W. (2004) *Copula Methods in Finance*, John Wiley, New York.

Duffie, D. and Singleton, K. J. (2003) *Credit Risk*, Princeton University Press, Princeton and Oxford.

Joe, H. (1997) *Multivariate Models and Dependence Concepts*, Chapman & Hall, London.

Li, D (2000) On default correlation: A copula function approach, *Journal of Fixed Income*, **9**, 43–54.

Mari, D. D. and Kotz, S. (2001) *Correlation and Dependence*, World Scientific, London.

McNeil, A., Frey, R., and Embrechts, P. (2005) *Quantitative Risk Management*, Princeton University Press, Princeton and Oxford.

Nelsen, R. B. (2007) *An Introduction to Copulas,* 2nd ed., Springer, New York.

Salmon, F. (2009) Recipe for Disaster: The Formula That Killed Wall Street, *Wired* http://www.wired.com/techbiz/it/magazine/17-03/wp_quant?currentPage=all