University of Göttingen

copR:
Multivariate Probability Distributions with Archimedean Copulas in R

Course: Statistical Programming with R

Students:

Alexander Ritz,
Student ID: 21800461

Juliane Wilcke,
Student ID: 21742610

Supervisor:
Rouven Schmidt, M.Sc.

23.09.2019

Contents

1 Introduction

The ability to accurately model multivariate distributions is of utmost importance in countless applications of statistical theory. Straightforward examples of such applications are the evaluation of financial portfolios or the examination of disease occurence in some population. The random variables of interest are usually collected into a random vector in such cases. Such a random vector then represents the multivariate distribution of interest.
Whenever the distribution of such random vectors is of interest, the dependence structure defining the relationship between the individual random variables becomes a focal point of statistical analysis, and rightfully so. Unfortunately, in the past the construction of multivariate distributions was rather limited. The usual approach of generalising a univariate distribution to multiple dimensions leaves little room for flexibility. In many cases, the individual random variables have to adhere to identical distributions. Even without that particular restriction, oftentimes the form of dependence is similarly restricted [Hofert et al. 2018].

Copulas offer a more flexible approach in comparison. Constructing a multivariate probability distribution with the help of copulas opens up the possibility of choosing the marginal distributions and the dependence structure individually. This paper will give a short introduction to the relevant theory relating to a specific class of copulas, so called *Archimedean* copulas, and showcase the R package *copR*. copR offers basic functionality regarding the construction, sampling, visualisation and fitting of Archimedean copulas. The first half of this text will focus on the necessary mathematical concepts, leaving the more intricate details to the appendix, while the latter half will give an overview of copR's functionality and structure. Some more involved details of the implementation are detailed in the appendix.

2 Theory of Archimedean Copulas

2.1 What is a Copula?

When analysing the joint distribution of several random variables, two concepts are of particular interest: The individual univariate distributions of the involved variables, the margins, and the dependence structure among the margins. The term copula [1] derives from the fact that the joint distribution and the margins are **linked** through a copula, the copula taking the place of the isolated dependence structure.

The following example demonstrates the basic approach of what a copula is supposed to offer the practising statistician. Imagine an arbitrary continuous random vector $\mathbf{X}$, with

$$\mathbf{X} = (X_1, X_2) \sim F, \text{ with } X_1 \sim F_1 \text{ and } X_2 \sim F_2$$

An intuitive way to strip this distribution of any information not immediately related to the dependence structure would be the application of a well known transformation [Hofert et al. 2018].

Definition 1 *The Probability Transformation*

$$\textit{If } X \sim F \textit{ and } F \textit{ is a continuous cdf, then:} \quad U = F(X) \sim \mathcal{U}(0,1)$$

$U = F(X)$ is often called the grade of X.

The related Quantile transformation obviously serves as an inversion of the transformation above:

Definition 2 *The Quantile Transformation*

$$\textit{If } U \sim \mathcal{U}(0,1) \textit{ and } F \textit{ is a cdf, then:} \quad X = F^{-1}(U) \sim F$$

In order to decompose the joint distribution into the marginal distributions and the information on their dependence, the random variables X_1 and X_2 are transformed into their grades U_1 and U_2. The resultant joint distibution of the grades can then be interpreted as the dependence structure stripped of the information exclusively relating to the marginals [Meucci 2011].

The following graphs illustrate the approach based on 500 simulated observations of three multivariate normal distributions, each with a different form of dependence.

[1] From the latin word *copula*, meaning connection.

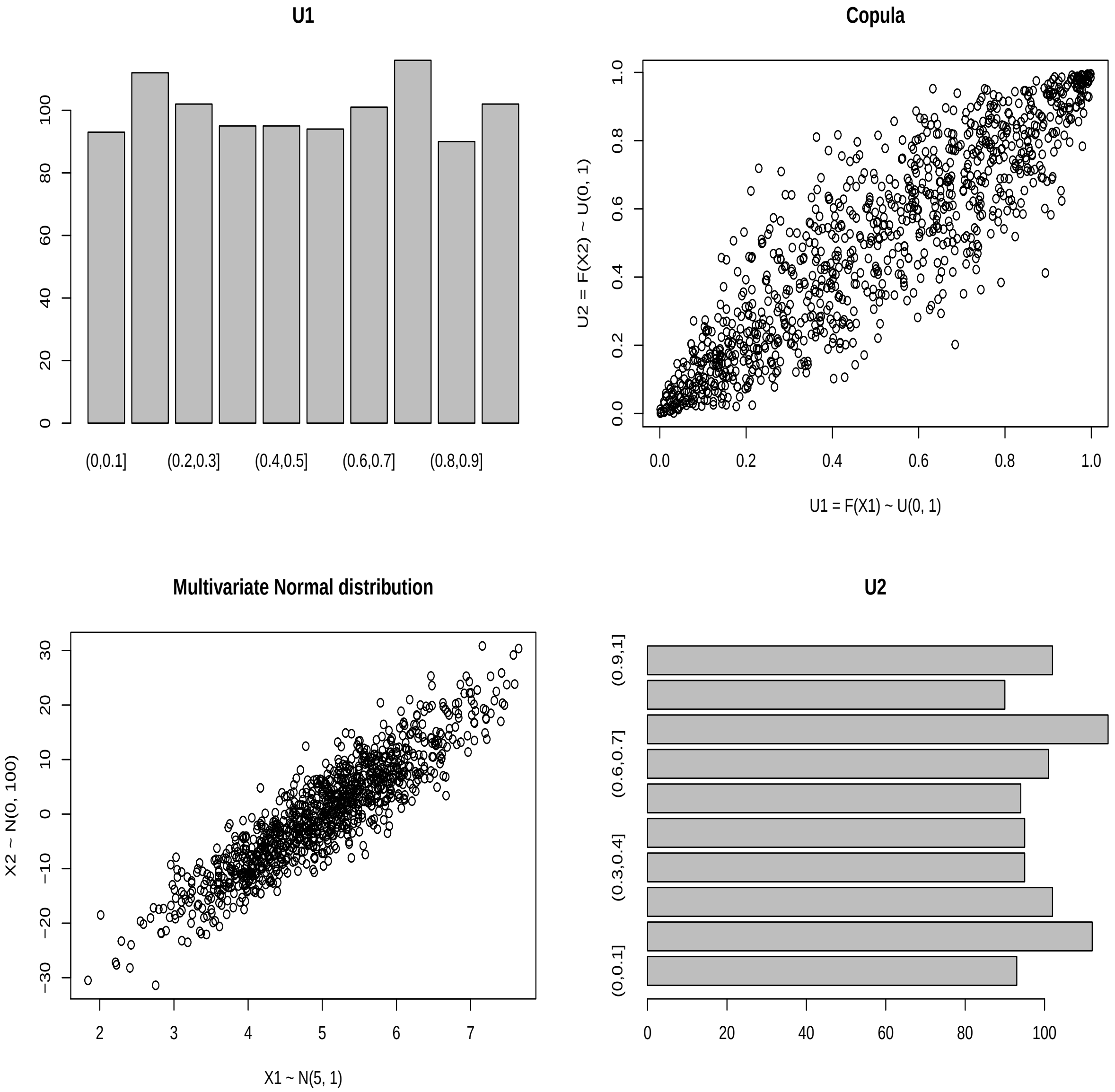


Figure 1: Copula construction illustrated for positive correlation

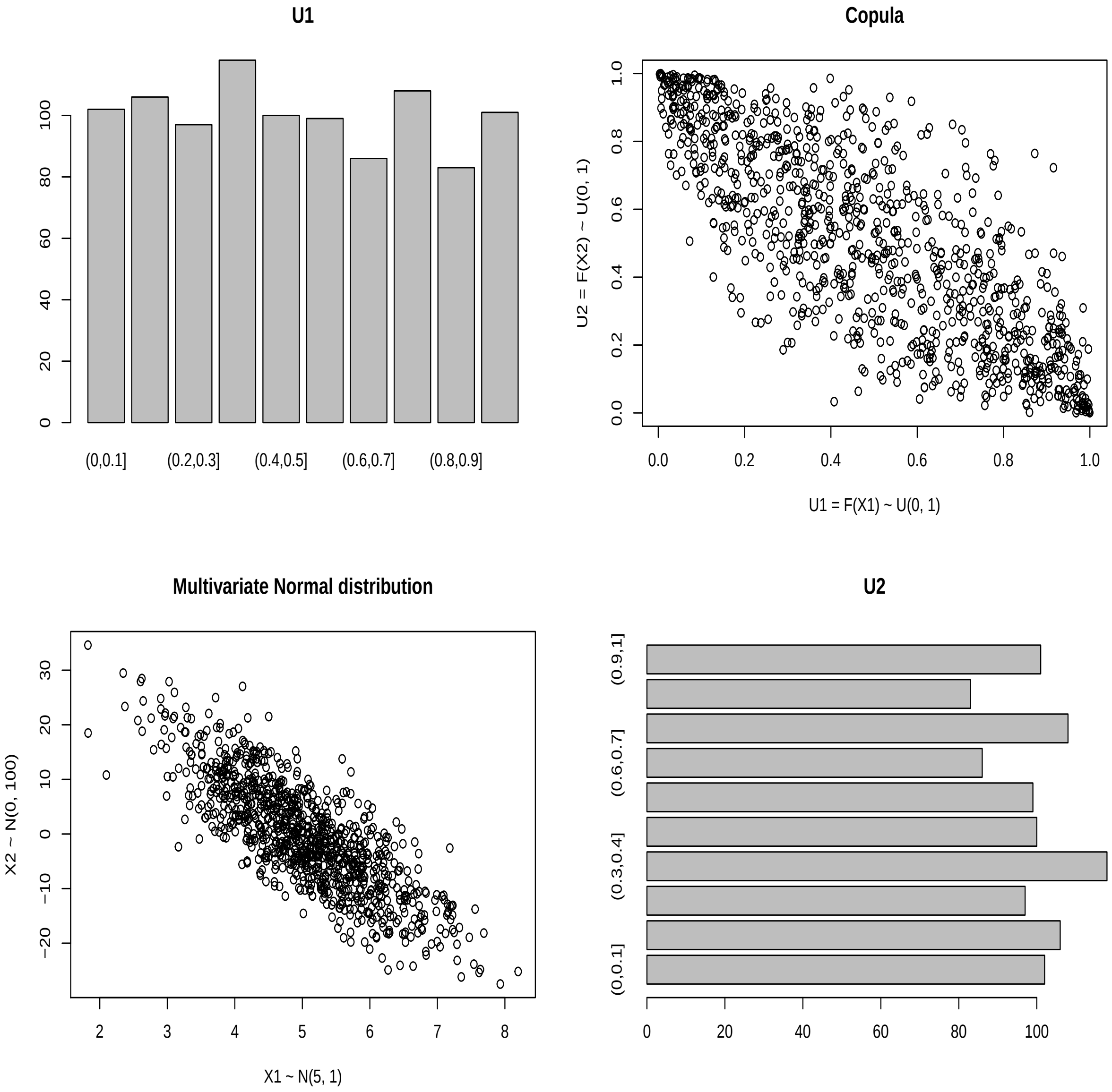


Figure 2: Copula construction illustrated for negative correlation

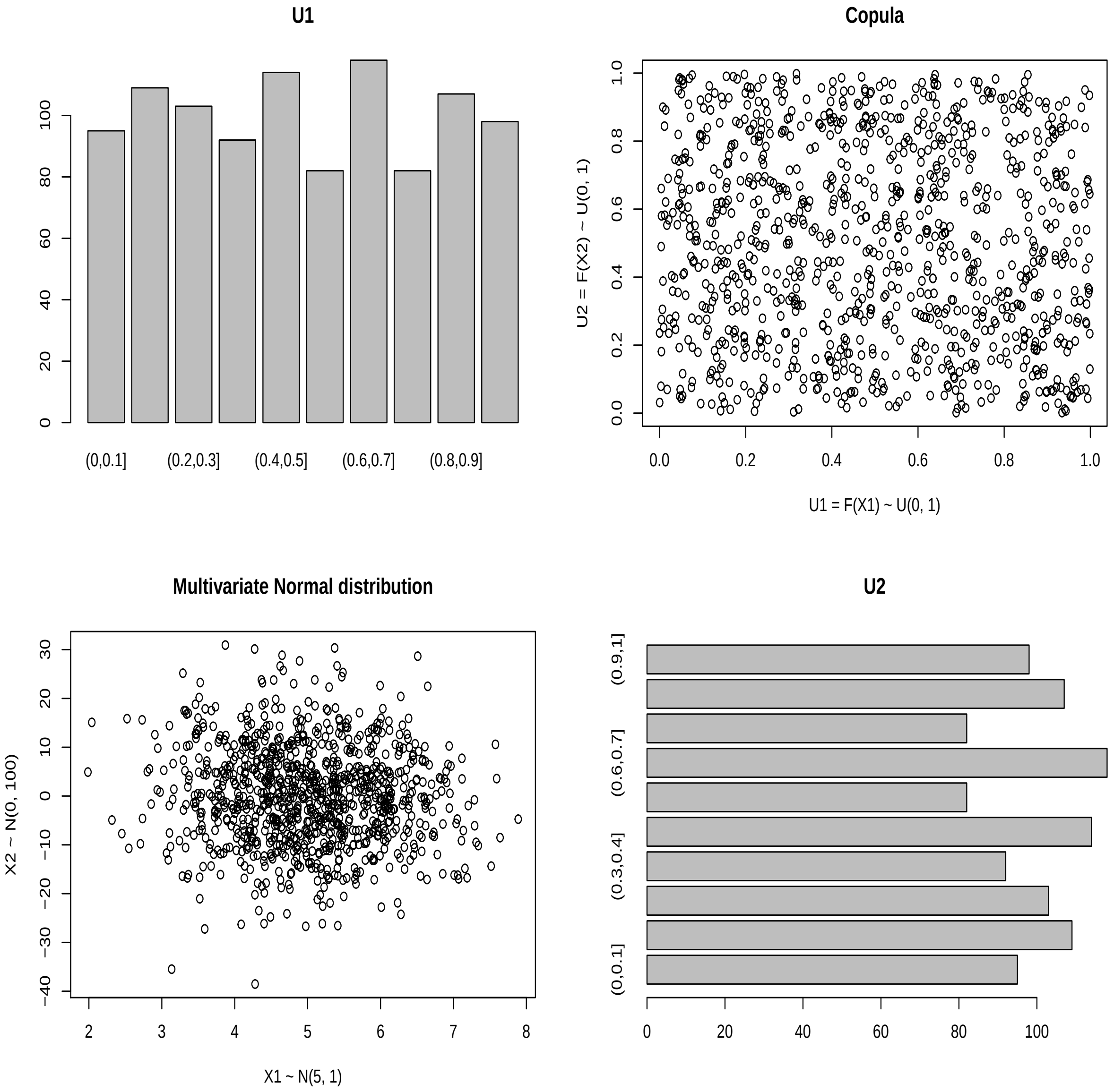


Figure 3: Copula construction illustrated for independence

2.2 Formal Definition and Sklar's Theorem

Following the preceding thoughts, it is only natural to define a copula formally as the joint distribution of the grades. In the bivariate case the properties are easier to parse and interpret than for an arbitrary number of dimensions [Nelsen 2006]:

Definition 3 *A bivariate copula is a function*

$$C : [0,1]^2 \to [0,1] :$$

$$(u_1, u_2) \mapsto C(u_1, u_2)$$

such that:

- $\forall u \in [0,1]^2 : \quad C(u_1, 0) = C(0, u_2) = 0$
- $C(u_1, 1) = u_1$ *and* $C(1, u_2) = u_2$
- *C is 2-increasing, meaning:* $C(u_1, u_2) - C(u_1, v_2) - C(v_1, u_2) + C(v_1, v_2) \geq 0$, *for* $v_1 \leq u_1$ *and* $v_2 \leq u_2$

It is relatively obvious that the properties ensure that C represents an 2-dimensional cumulative distribution function with unifrom margins.

Having defined what a copula is, one question remains unanswered: What is the exact relationship between copula, joint distribution and marginal distributions? Equating the different distributions with their respective distribution function and the insights contained in *Sklar's Theorem* offer an answer [Hofert et al. 2018].

Definition 4 *The bivariate form of Sklar's Theorem*
Let $F_{\boldsymbol{X}}$ *be a bivariate joint cdf for the RV* $\boldsymbol{X} = (X_1, X_2)$ *with marginal cdfs* F_{X_1} *and* F_{X_2}*. Then there exists a copula C such that:*

$$F_{\boldsymbol{X}}(x_1, x_2) = C(F_{X_1}(x_1), F_{X_2}(x_2)) = C(u_1, u_2),$$

with $x_1, x_2 \in \bar{\mathbb{R}}$ *and* $\bar{\mathbb{R}} := \mathbb{R} \cup \{-\infty, \infty\}$

Assuming $F_{\mathbf{X}}$ and C are differentiable, calculating the joint density in combination with Sklar's Theorem gives [Meucci 2011];[Hofert et al. 2018]:

$$\frac{\partial^2 F_{\mathbf{X}}(x_1, x_2)}{\partial x_1 \partial x_2} = f_{\mathbf{X}}(x_1, x_2) = c(F_{X_1}(x_1), F_{X_2}(x_2)) f_{X_1}(x_1) f_{X_2}(x_2)$$

with

$$c(F_{X_1}(x_1), F_{X_2}(x_2)) = \frac{\partial^2 C(F_{X_1}(x_1), F_{X_2}(x_2))}{\partial F_{X_1}(x_1) \partial F_{X_2}(x_2)},$$

The copula density can therefore be interpreted as a multiplicative adjustment of the joint density under independence (that is: $f_{X_1}(x_1) f_{X_2}(x_2)$). The concept transfers to higher dimensions in an intuitive and analogous manner.

2.3 Archimedean Copulas

One particular class of copulas that can be considered are Archimedean copulas. A copula C is called an Archimedean copula if it can be written in terms of a continuous generator function $\phi : [0, \infty] \to [0, 1]$ like so [Hofert et al. 2018]:

$$C(u) = \phi(\phi^{-1}(u_1) + ... + \phi^{-1}(u_n)),$$

with $u \in [0, 1]^n$.

The generator function ϕ holds the properties:

- $\phi(0) = 1$
- $\phi(\infty) = 0$
- $(-1)^k \phi^{(k)}(t) \geq 0$, for $k \in \{0, 1, ..., n-2\}, t \in (0, \infty)$

Archimedean copulas can be classified in a natural manner based on the form of this generator function. Examples of generators and their corresponding family names are listed in Table 1 [Hofert et al. 2018].

[2] $\theta \in \mathbb{R} \backslash 0$ in the bivariate case.

Family	$\phi(\theta)$	Parameter space
Clayton	$\phi(t) = (1+t)^{-\frac{1}{\theta}}$	$\theta \in (0, \infty)$
Frank	$\phi(t) = -\log\left(1 - (1 - \exp(-\theta)) \exp(-t)\right)/\theta$	$\theta \in (0, \infty)^2$
Gumbel-Hougaard	$\phi(t) = \exp(-t^{\frac{1}{\theta}})$	$\theta \in [1, \infty)$
Ali-Mikhail-Haq	$\phi(t) = (1-\theta)/\exp(t) - \theta$	$\theta \in [0, 1)$

Table 1: Examples of Archimedean families and their generating functions.

3 Implementation

The package copR provides functionality for constructing Archimedean copulas, which can then be sampled from, visualised, and fitted to data. Two families of Archimedean copulas have been implemented, the Clayton copula family and the Frank copula family. The Independence copula[3] has also been implemented. The package is structured in a way that enables the implementation of further families without unnecessary difficulties.

3.1 Construction

The functions claycop, frankcop and indcop can be used to construct a Clayton copula object, a Frank copula object, or an Independence copula object, respectively. The number of dimensions and, unless Independence is chosen, a parameter value need to be supplied. Each constructor function returns a list of the corresponding S3 class, containing the following elements: the number of dimensions, expressions for the generator function and its inverse, the parameter value of the generator function, the upper and lower bounds of permitted parameter values, the name of the copula family, and expressions for the density (pdf) and the distribution function (cdf) of the copula. The Independence copula does not have any parameters and therefore also no paramter bounds. Although possible to define, a generator function has not been implemented for the Independence copula due to a lack of utility.

The expressions for the density and distribution functions are constructed by a call from within the constructor functions to the generic functions dcop and pcop, which have three methods each for the three classes of copula implemented. The three copula classes belong to a common S3 superclass called copula and inherit

[3] The Independence copula can be interpreted as an Archimedean copula, albeit without any parameter.

the latter's methods for the generic functions `print` and `summary`. Whether a given copula object belongs to one of these four classes may be checked with the four corresponding `is.classname` functions.

3.2 Sampling

In order to obtain a random sample from a given Archimedean copula, the copula object and the number of desired observations need to be supplied to the generic function `rcop`, which also relies on three methods for the types of copula implemented. The sampling of Clayton and Frank copulas is based on the algorithm of Marshall and Olkin (1988) for dimensions higher than 2 and the conditional approach otherwise. It makes use of an internal function called `invgeneval` to evaluate the appropriate inverse generator function. `rcop` returns a matrix containing the desired number of observations (rows) of the random variables.

3.3 Visualisation

The density and distribution functions of all implemented Archimedean copula types can be visualised in contour plots using the function `coptourplot` and in three-dimensional perspective plots using the function `coppersplot`. Common code for constructing the xy-grid and evaluating the function of interest at each point of the grid has been moved to the internal function `zofgrid`. While reasonable defaults have been chosen, a large number of plotting options can be adjusted by the user, including the colour palette for contours and facets, which may be chosen to take colour blindness into account or to be optimised for grayscale printing.

3.4 Fitting

Given an Archimedean copula object along with data, the function `cloglik` computes the value of the log-likelihood function using the aforementioned function `dcop` for the density. The function gives the option of supplying a parameter value differing from the parameter of the supplied copula. Using the function `cfit`, the copula parameter can be fit to the data by determining the maximum of the log-likelihood function computed by `cloglik`. `cfit` returns a list of the S3 class `fitcop`, which inherits the class of the supplied copula object and has identical elements to the original object, with three exceptions: The original parameter value gets replaced by the maximum likelihood estimate, and two additional elements contain the maximum value of the log-likelihood and the number of observations that were used, respectively. For model selection purposes, both the AIC and BIC have been implemented to be applied to a `fitcop` object. The functions are called `c_aic` and `c_bic` .

The parameter of a given bivariate Archimedean copula can also be fitted to the data using the method of moments, in this case based on Kendall's tau. The wrapper function `corken` computes all pairwise Kendall rank correlation coefficients between the variables of a given data set. A fast option is available for large data sets without missing values. If Kendall's tau is supplied to the function `cmethmo`, along with a copula object, that tau is used to estimate the new parameter. Alternatively a data set can be supplied to calculate Kendall's tau. `cmethmo` returns a copula object with the parameter determined by the inversion of Kendall's tau.

4 Usage

For a start, the package `copR` is installed from GitHub und loaded. Note that the package was created in R version 3.6.0 and, importantly, has no dependencies other than the `base` package and three packages loaded on startup by default (i.e. `grDevices`, `graphics`, `stats`).

```
if (!require(devtools))
  install.packages(devtools)
devtools::install_github("AlexanderRitz/copR")
library(copR)
```

4.1 Construction

Using `copR` one can now construct Archimedean copulas of type Clayton, Frank, and Independence in, for example, two dimensions.

```
ccop2 <- claycop(par = 5, dim = 2)
fcop2 <- frankcop(par = 3, dim = 2)
icop2 <- indcop(dim = 2)
```

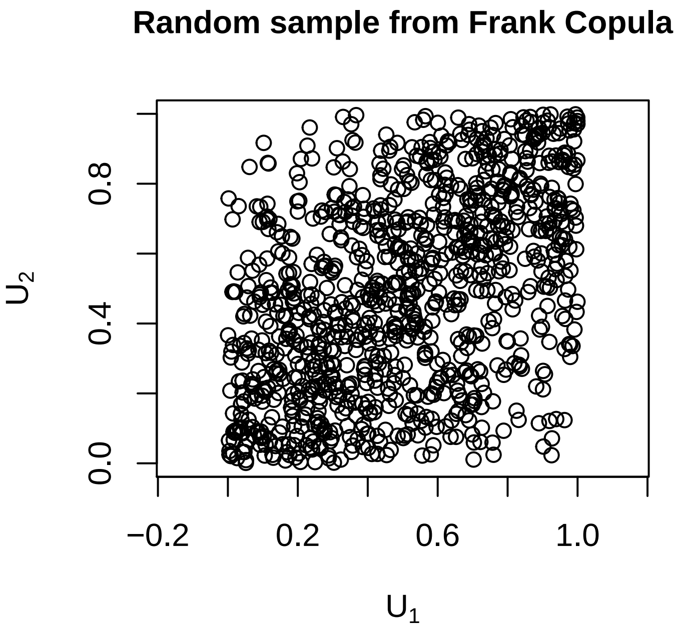
Printing and summary methods enable the viewer to get an overview of a copula.

```
ccop2

## A Clayton copula in 2 dimensions with parameter theta = 5.
##
##  pdf omitted due to length, access with "yourcopula"$distribution$cdf
##  pdf omitted due to length, access with "yourcopula"$distribution$pdf

summary(ccop2)

## Clayton copula:
##  Dimensions:  2
##  Parameter value:  5
##  Parameter range:  (0, Inf)
##  Generator function:  t^(-theta) - 1
##  Inverse generator:  (1 + s)^(-(1/theta))
```

4.2 Sampling

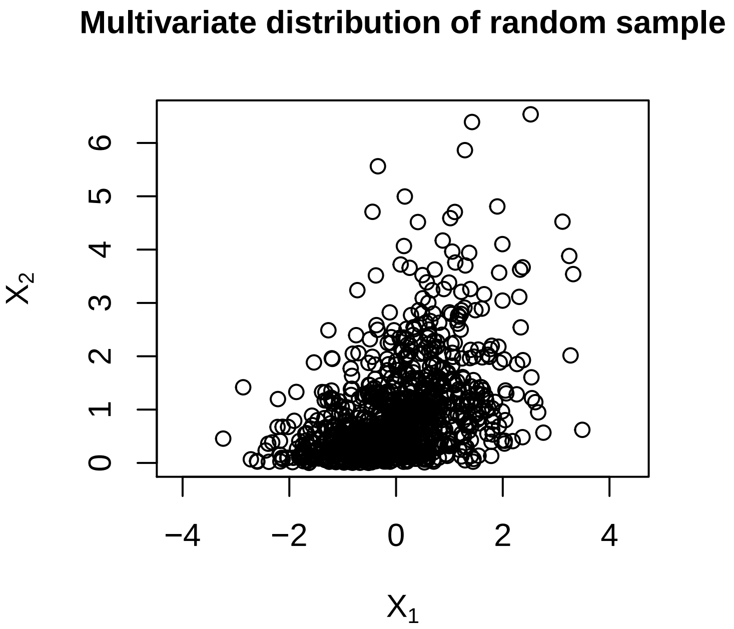
One may want to create a random sample with 1000 data points from the contructed Frank copula and plot it.

```
set.seed(239)
U2 <- rcop(copula = fcop2, n = 1000)
plot(U2, asp = 1, xlab = quote(U[1]), ylab = quote(U[2]),
     main = "Random sample from Frank Copula", cex.main = 1)
```



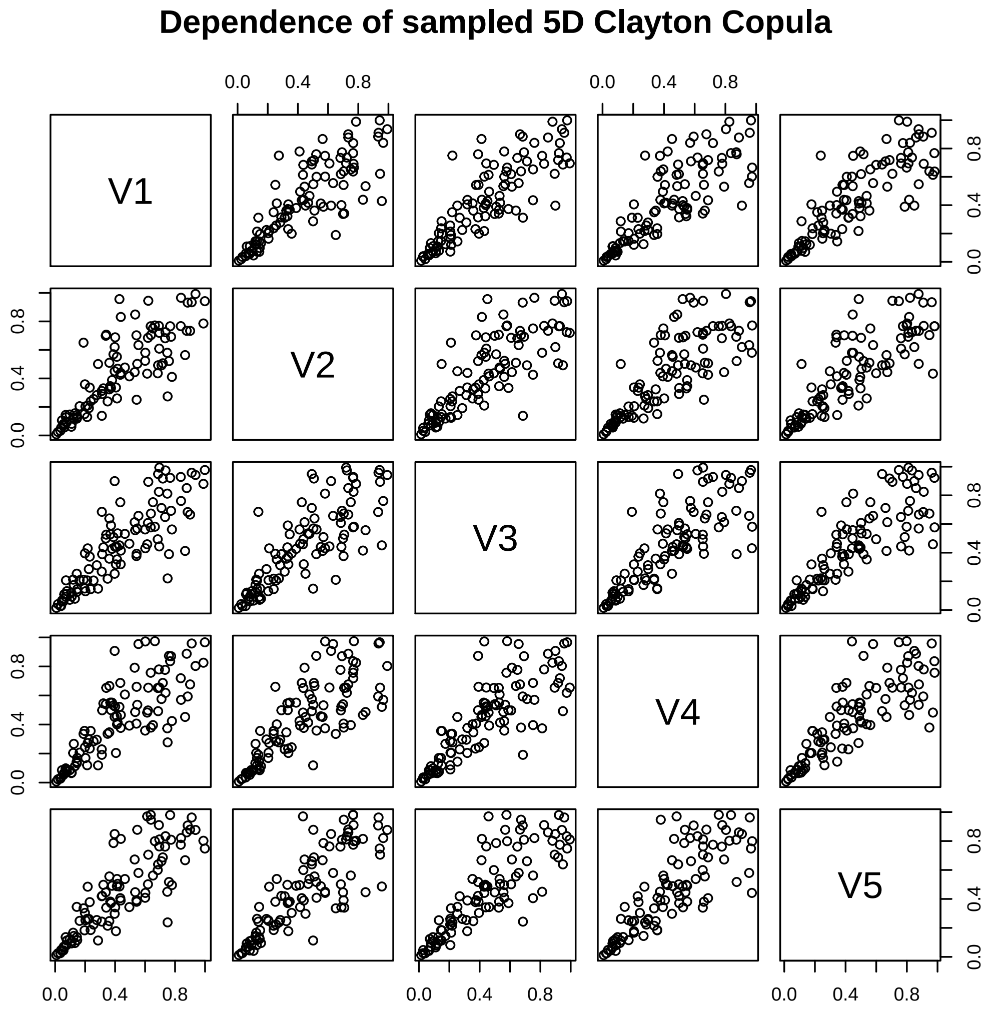
If one assumes that the first marginal is normally distributed and the second one exponentially, one can quantile transform the sample and plot its multivariate distribution.

```
X_1 <- qnorm(U2[, 1], 0, 1)
X_2 <- qexp(U2[, 2], 1)
plot(X_1, X_2, asp = 1, xlab = quote(X[1]), ylab = quote(X[2]),
     main = "Multivariate distribution of random sample",
     cex.main = 1)
```



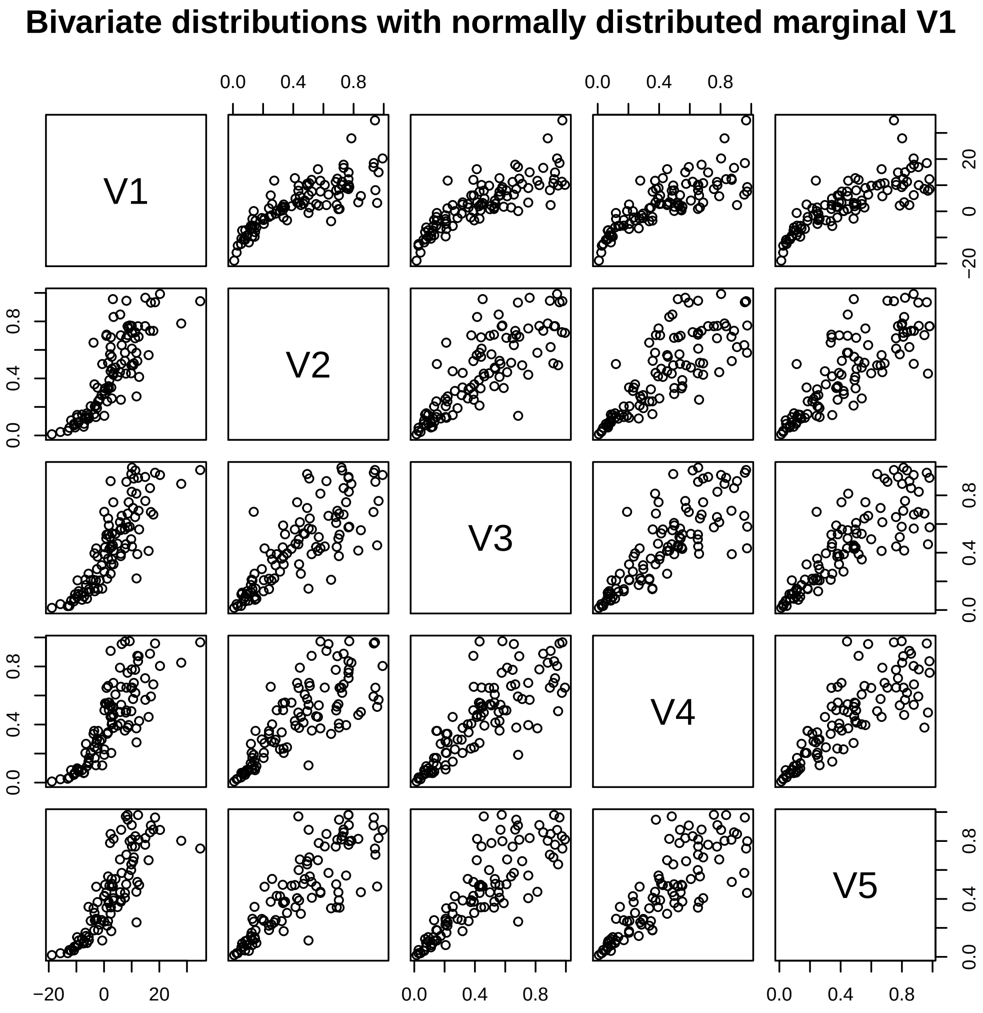
One may also look at copulas of higher dimensions, say 5D, and examine the dependence structure of the marginals among each other.

```r
ccop5 <- claycop(par = 4, dim = 5)
U5 <- rcop(copula = ccop5, n = 100)
U5 <- as.data.frame(U5)
pairs(~ V1 + V2 + V2 + V3 + V4 + V5, data = U5, main =
      "Dependence of sampled 5D Clayton Copula")
```



Again, one can consider what happens when the distribution of the first variable is transformed into a normally distributed variable.

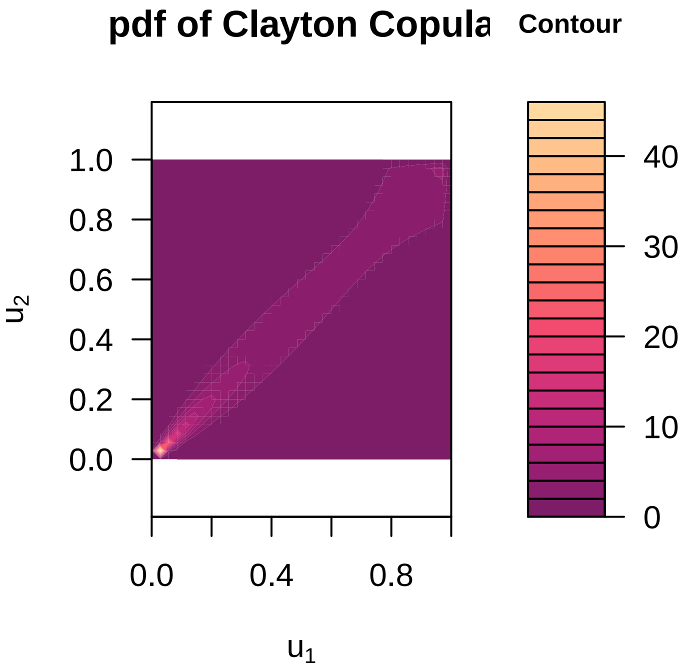
```r
U5$V1 <- qnorm(U5$V1, mean = 5, sd = 10)
pairs(~ V1 + V2 + V2 + V3 + V4 + V5, data = U5,
      main = paste("Bivariate distributions with normally",
                   "distributed marginal V1", sep = " "))
```



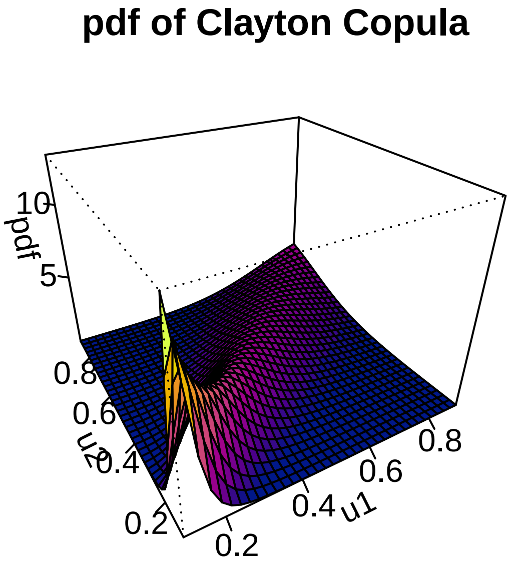
4.3 Visualisation

The density and distribution function of a constructed bivariate copula object may be plotted in two or three dimensions. Consider the density of the 2D Clayton copula as an illustration how some of the display options can be adjusted :

```
coptourplot(copula = ccop2, FUN = "pdf", n.grid = 36, nlevels = 30,
                   col.pal = "SunsetDark", con.lines = FALSE)
```

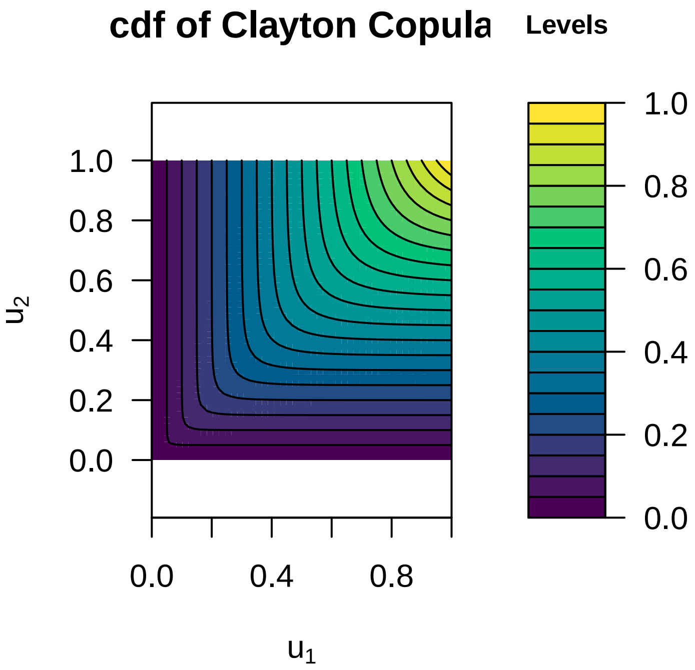


```
coppersplot(copula = ccop2, FUN = "pdf", n.grid = 36, delta= 0.1,
            theta = -30, col.pal = "Plasma", border = NULL)
```

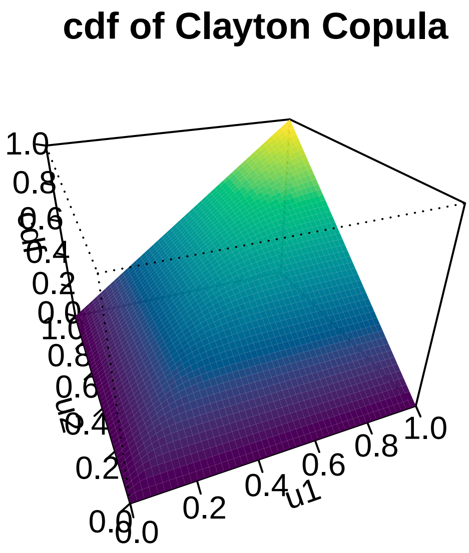


Going with the preset options to visualise the distributon function:

```
coptourplot(copula = ccop2, FUN = "cdf")
```



```
coppersplot(copula = ccop2, FUN = "cdf")
```



4.4 Fitting

Given some data, one may calculate the value of the log-likelihood function associated with a Clayton copula.

```
U <- matrix(c(0.7026, 0.6359, 0.4116, 0.8833, 0.3127, 0.4035),
            nrow = 3, byrow = TRUE)
cloglik(copula = ccop2, data = U)

## [1] -0.2877945
```

The output of the `cfit` function gives the maximum likelihood estimate of the parameter value ("theta") und also the maximum value of the log-likelihood function ("maximum").

```
cfit(copula = ccop2, data = U)

## The 2 dimensional Clayton copula has been fitted with theta = 2.02622656753224.
##
## The maximum value of the log-likelihood function was 0.441663538024537.
```

Finally, estimating the parameter of a copula based on the inversion of Kendall's tau given some data:

```
cmethmo(copula = ccop2, data = U)

## A Clayton copula in 2 dimensions with parameter theta = 1.
##
##  pdf omitted due to length, access with "yourcopula"$distribution$cdf
##  pdf omitted due to length, access with "yourcopula"$distribution$pdf
```

A Appendix

A.1 General copula definition

Compare [Nelsen 2006]

Definition 5 *A copula is a function*

$$C : [0,1]^n \to [0,1] :$$

$$(u_1, ..., u_n) \mapsto C(u_1, ..., u_n),$$

such that:

- $\forall u \in [0,1]^n : \quad \exists u_j = 0 : \quad C(u) = 0, \quad j = 1, ..., n$
- $\forall u_j : \qquad C(1, ..., u_j, ..., 1) = u_j \quad (\textit{that is: } (u_1, ..., u_{j-1}, u_{j+1}, ...u_n = 1)$
- *C is n-increasing*

A.1.1 n-inreasing functions

Compare [Nelsen 2006]

Definition 6 *A function $C : [0,1]^n \to [0,1]$ is n-increasing if for any hyper-rectangle*

$$R = \prod_{i=1}^{n} [x_i, y_i] \subseteq [0,1]^n$$

the C-measure / C-volume is non-negative:

$$V_C(R) \geq 0$$

A.1.2 Fréchet-Hoeffding bounds

Compare [Nelsen 2006]

Definition 7 *For a bivariate copula, the following bounds are called the Fréchet-Hoeffding bounds:*

$$max(u_1 + u_2 - 1, 0) \leq C(u_1, u_2) \leq min(u_1, u_2) \tag{1}$$

The upper bound is always (not just in the bivariate case) a copula itself. The lower boundary only in the case of two variables.

A.2 Properties of Archimedean copulas

The class of Archimedean copulas is an associative class of copulas, i.e.:

Definition 8

$$C(C(u_1, u_2), u_3) = C(u_1, C(u_2, u_3))$$

The class of Archimedean copulas is a permutation-symmetric class of copulas, i.e.:

Definition 9

$$C(u_1, u_2, u_3) = C(u_3, u_2, u_1)$$

A.3 Details of implementation

A.3.1 Sampling algorithms

The following paragraphs give a short overview of the algorithms used to simulate copula samples within copR.

A d-dimensional Clayton copula is sampled by utilising the Marshall-Olkin algorithm for Archimedean copulas generated by completely monotone ϕ [Marshall and Olkin 1988];[Hofert et al. 2018]. The algorithm involves four steps:

- Simulate d independent uniform variable u_i for $i = 1, ..., d$
- Simulate a variable Y which is Gamma distributed with parameters $(\frac{1}{\theta}, 1)$

- Calculate $s_i = \frac{-ln(u_i)}{Y}$ for $i = 1, ..., d$
- Then $X_i = \phi^{-1}(s_i)$ is a realization of a Clayton copula.

A 2-dimensional Frank copula is sampled with the conditional approach [compare Hofert et al. 2018 and Whelan 2004]. A d-dimensional Frank copula can be sampled analogously to the Clayton. Although one has to simulate a Logarithmic distribution in the second step insted of Gamma. This is achieved by *Kemp's second accelerated generator* (Devroye 1986, p. 548)

The independence copula can be trivially implemented with the help of *runif*.

A.4 Estimation

Maximum likelihood estimation is implemented in a framework equivalent to [Hofert et al. 2012], while the method of moments is implemented based on the relationships between Kendall's tau and the copula parameter theta as they are found in [Frees & Valdez 1998]. In the case of the Frank copula this relationship involves the Debye function [Genest 1987], which has therefore been implemented for order 1.

References

[Devroye 1986] Devroye, Luc :Non-Uniform Random Variate Generation. Springer 1986

[Frees & Valdez 1998] Fres, Edward W.; Valdez, Emiliano A. : Understanding Relationships Using Copulas. North American Actuarial Journal 2(1):1-25.

[Genest 1987] Genest, Christian. : Frank's Family of Bivariate Distributions. Biometrika, Vol. 74, No. 3, pp. 549-555

[Hofert et al. 2018] Hofert, Marius; Kojadinovic, Ivan; Mächler, Martin; Jun, Yan. : Elements of Copula Modeling with R. Springer 2018

[Hofert et al. 2012] Hofert, Marius; Mächler, Martin; McNeil, Alexander J.. : Likelihood inference for Archimedean copulas in high dimensions under known margins. Journal of Multivariate Analysis 110 (2012) 133–150

[Marshall & Olkin 1988] Marshall, Albert W.; Olkin, Ingram : Families of Multivariate Distributions. In: Journal of the American Statistical Association, Vol. 83, No. 403, pp.834-841

[Meucci 2011] Meucci, Attilio : Short, Comprehensive, Practical Guide to Copulas. GARP Risk Professional, October 2011, p. 22-27

[Nelsen 2006] Nelsen, Roger B. : An Introduction to Copulas. Second Edition. Springer 2006

[Whelan 2004] Whelan, Niall : Sampling from Archimedean copulas. Quantitative Finance, Volume 4, Issue 3, pp. 339-352 (2004)