Introduction to Copulas and Applications in Finance

Code

Assoc Prof Peter Julian Cayton, PhD

Member, Mathematical Statistics Research Group

Chair, Forecasting, Econometrics, and Time Series Research Group

School of Statistics

University of the Philippines Diliman

Google Scholar (https://scholar.google.com/citations?user=KtH_mGEAAAAJ), LinkedIn (https://www.linkedin.com/in/peter-julian-cayton-90a0a524/), X (formerly Twitter) (https://twitter.com/PJACaytonPhD)

Hide

```
### Preamble Code in Preparation

# install.packages("qrmtools")
# ## Quantitative Risk Management Tools package for financial econometrics tools
# install.packages("qrmdata")
# ## Quantitative Risk Management Data package for sample datasets
# install.packages("fpp2")
# ## Forecasting: Principles and Practice package for time series tools
# install.packages("tidyverse")
# ## Tidyverse Packages for ease of data wrangling
# install.packages("xts")
# ## Extension for Time Series Objects package for data processing
# ## of financial time series
# install.packages("ADGofTest")
# ## Anderson-Darling Goodness of Fit Test
# install.packages("qqtest")
# ## Quantile-quantile Tests
# install.packages("copula")
# ## Copula Package
# install.packages("ismev")
# ## Introduction to Statistical Modeling of Extreme Values package for
# ## data and discussion on EV copulas
# install.packages("rugarch")

library(rugarch)
```

```
Warning: package ‘rugarch’ was built under R version 4.3.3
```

Hide

```
library(ismev)
```

```
Warning: package ‘ismev’ was built under R version 4.3.3
```

Hide

```
library(fpp2)
```

```
Warning: package ‘fpp2’ was built under R version 4.3.3Warning: package ‘forecast’ was built under R v
ersion 4.3.3Warning: package ‘fma’ was built under R version 4.3.3Warning: package ‘expsmooth’ was bui
lt under R version 4.3.3
```

Hide

```
library(tidyverse)
```

```
Warning: package ‘tidyverse’ was built under R version 4.3.3
```

Hide

```
library(xts)
```

```
Warning: package ‘xts’ was built under R version 4.3.3Warning: package ‘zoo’ was built under R version
4.3.2
```

Hide

```
library(ADGofTest)
library(qqtest)
```

```
Warning: package ‘qqtest’ was built under R version 4.3.3
```

Hide

```
library(copula)
```

```
Warning: package ‘copula’ was built under R version 4.3.3
```

Hide

```
library(qrmdata)
```

```
Warning: package ‘qrmdata’ was built under R version 4.3.3
```

Hide

```
library(qrmtools)
```

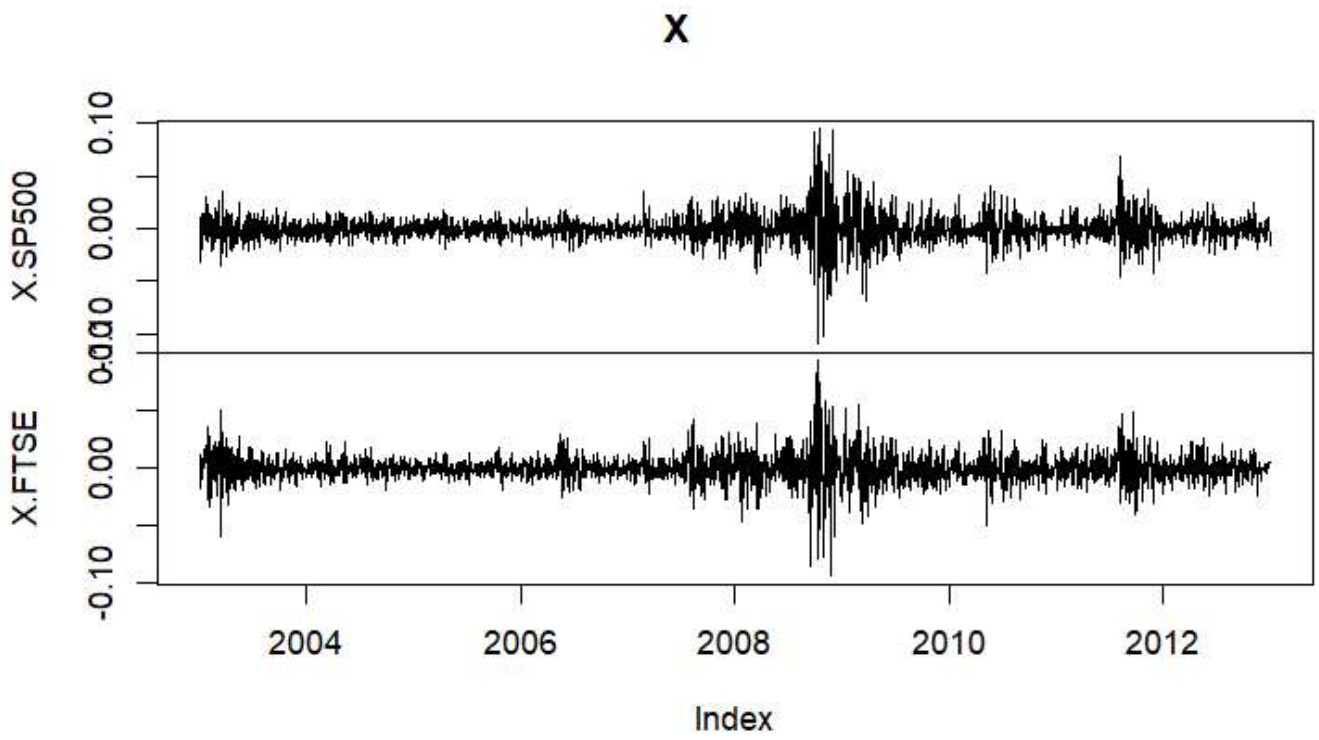
```
Warning: package ‘qrmtools’ was built under R version 4.3.3
```

A Motivating Example in Finance

Shown are the scatterplots for the daily percent changes, also called returns, of two stock market indices: S&P 500 and London FTSE.

Hide

```
## From the qrmdata package
## Source: https://github.com/qrmtutorial/qrm/blob/master/code/07_Copulas_and_Dependence/07_Copula_est
imation.R

## Load the time series
data("SP500")
data("FTSE")

## Build negative log-returns or losses
X.SP500 <- -returns(SP500)
X.FTSE  <- -returns(FTSE)

## Merge
X <- merge(X.SP500, X.FTSE, all = FALSE)
time <- c("2003-01-01", "2012-12-31")
X <- X[paste0(time, collapse = "/"),]

## Basic plot
plot.zoo(X)
```



Hide

```
## Observe that there are zero values caused by market closures

apply(X == 0, 2, sum)
```

```
X.SP500  X.FTSE
      2      51
```

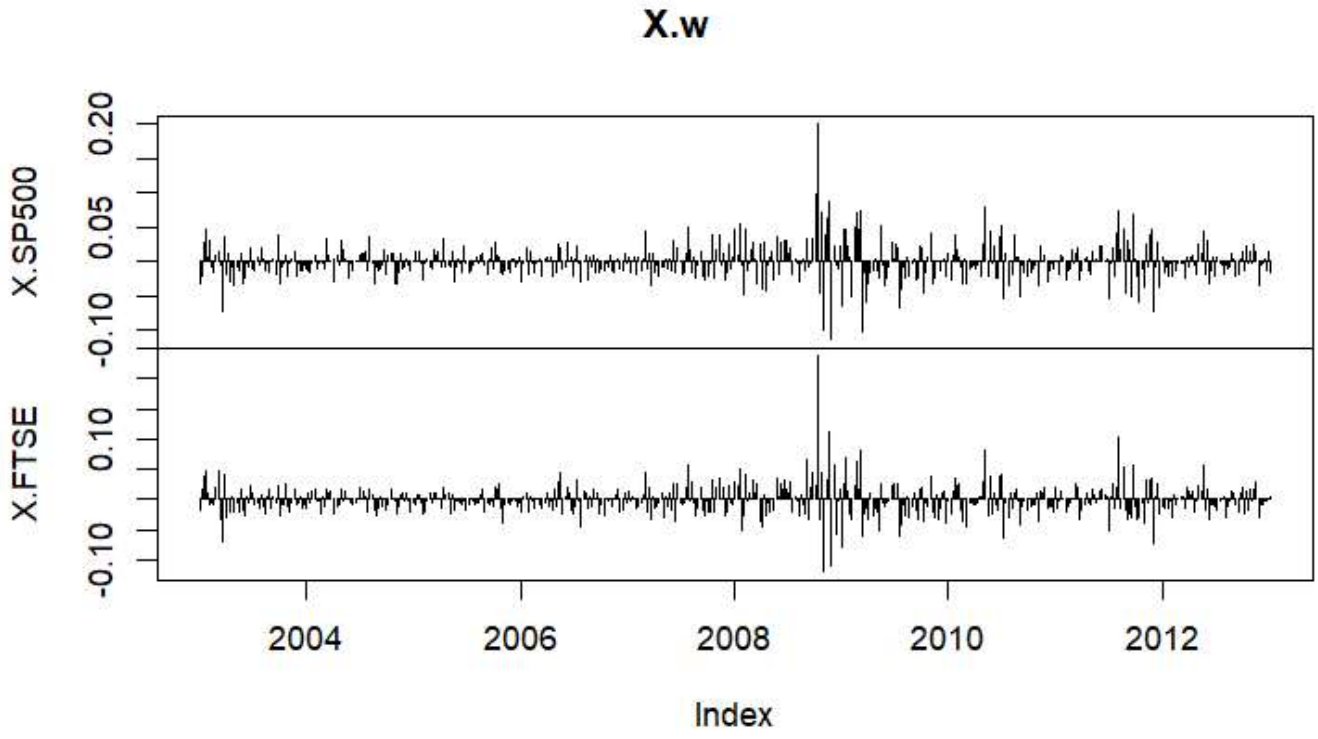
Hide

```
## Thus, remove data where zero returns from at least 1 is observed
rexcl <- (X[,1] == 0) | (X[,2] == 0)
X. <- X[!rexcl,]

## Aggregating by week
dim(X.w <- apply.weekly(X., FUN = colSums))
```

```
[1] 523   2
```

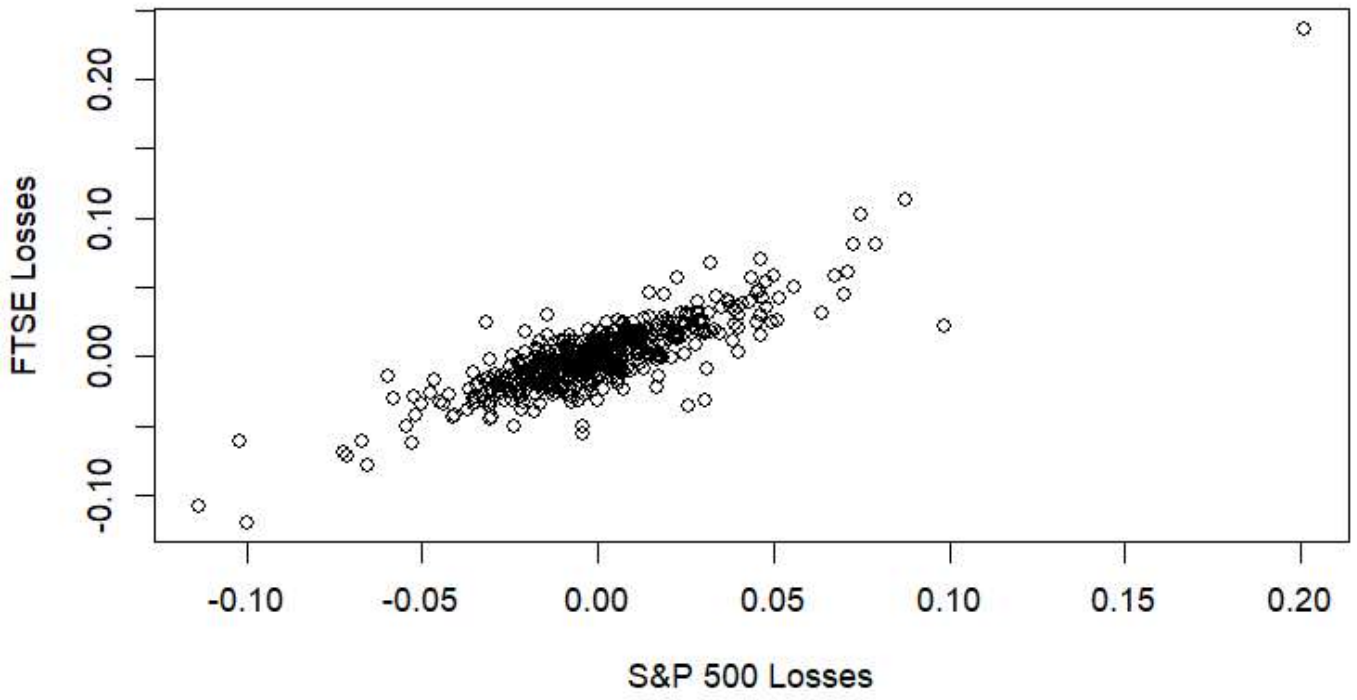
Hide

```
plot.zoo(X.w, type = "h")
```



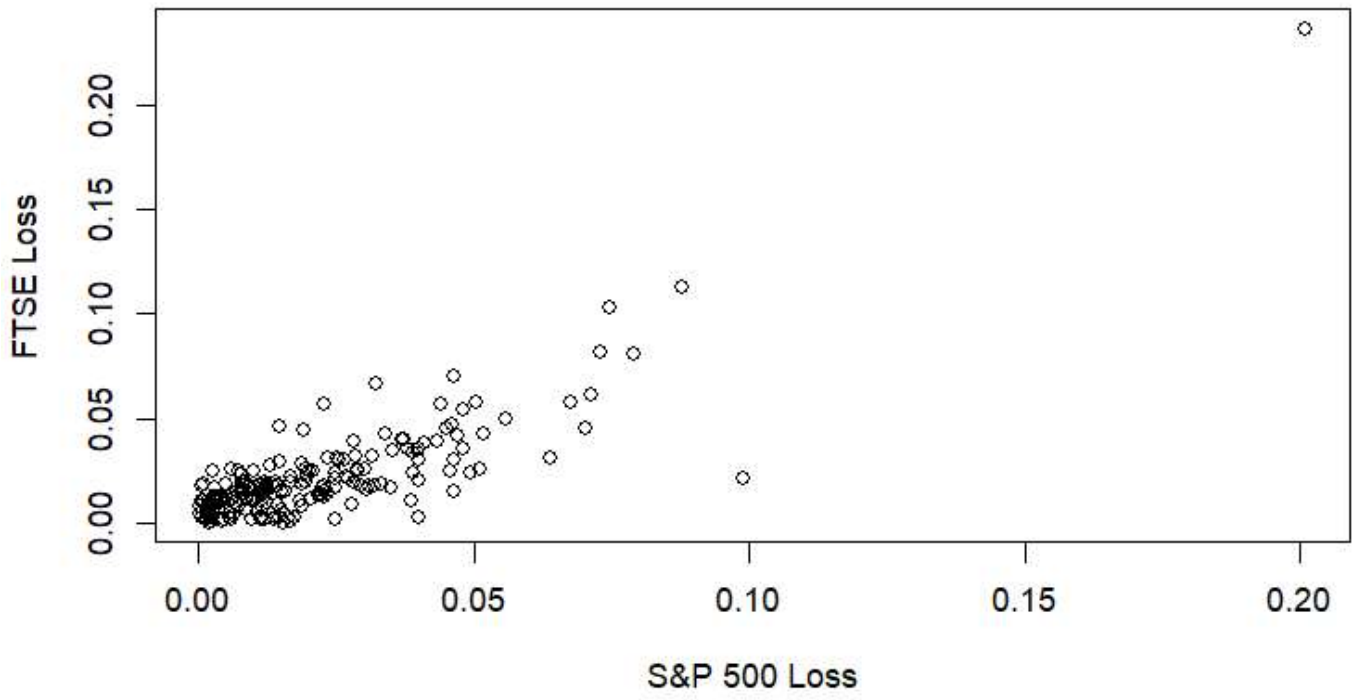
Hide

```
## Scatterplot
plot(as.matrix(X.w), xlab = "S&P 500 Losses", ylab = "FTSE Losses")
```



Hide

```
## Zoom into joint losses
Loss <- X.w
Linc <-  (Loss[,1] > 0) & (Loss[,2] > 0)
Loss.qrtr <- Loss[Linc,]

plot(as.matrix(Loss.qrtr), xlab = "S&P 500 Loss", ylab = "FTSE Loss")
```

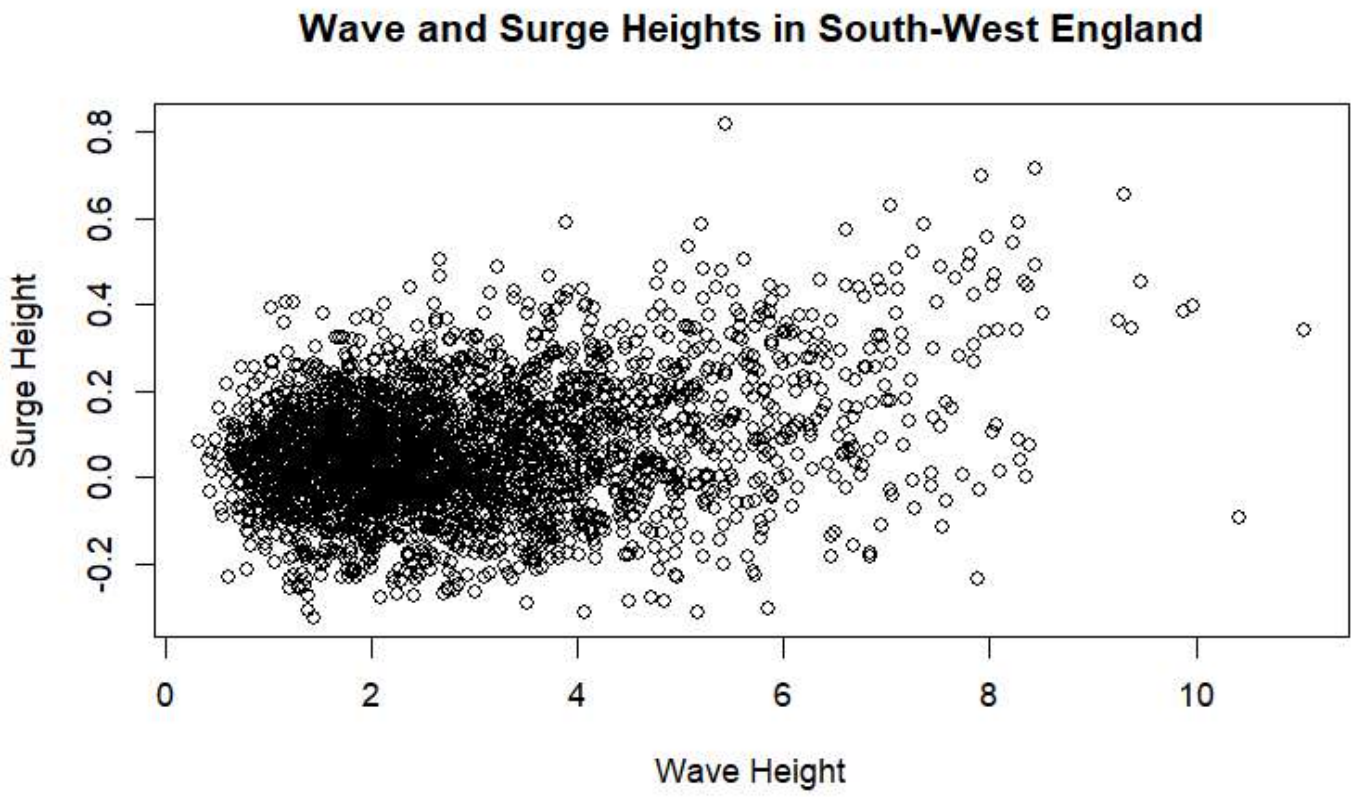


A Motivating Example in Environmental Modeling

The example below is from Coles (2001), shows the plot of wave and surge heights in South-West England.

Hide

```
data(wavesurge)

plot(wavesurge, xlab = "Wave Height", ylab = "Surge Height", main = "Wave and Surge Heights in South-W
est England")
```



Basic Concepts

In a multivariate probability course, we define a joint distribution $F_{X,Y}$ of random variables X and Y as:

$$F_{X,Y}(x,y) = \Pr(X \leq x, Y \leq y)$$

From the joint distribution, we can get a lot of information about X and Y and the association between the two. For example, we can extract the marginal distribution functions:

$$F_X(x) = \Pr(X \leq x) = F_{X,Y}(x, +\infty)$$

$$F_Y(y) = \Pr(Y \leq y) = F_{X,Y}(+\infty, y)$$

However, we cannot derive the joint distribution from the marginals unless we have additional information like some function C that helps in combining the marginals together,

$$F_{X,Y}(x,y) = C\left(F_X(x), F_Y(y)\right)$$

In a multivariate probability course in statistics, a common assumption for statistical modeling is *independence*, which implies the following structure:

$$F_{X,Y}(x,y) = F_X(x)F_Y(y) \quad \Rightarrow \quad C(u,v) = uv$$

But C can have a much more sophisticated structure that describe how to combine together marginal information to form joint distributions based on how variables are associated.

The C function that describes how marginal distribution of variables are combined together to form their joint distribution is called a **copula**.

Definition of a Copula

A function $C(u, v)$ is a (bivariate) *copula* such that $C : [0, 1]^2 \to [0, 1]$ and has the following features:

1. Boundedness: if at least 1 of the arguments in C is zero, then C is zero, and when all arguments are equal to one, then C is one.

$$C(0, v) = C(u, 0) = 0$$

$$C(1, 1) = 1$$

2. Monotone Nondecreasing: for two pairs of numbers $(a, b)'$ and $(c, d)'$ where $a \leq c$ and $b \leq d$, then

$$C(b, d) - C(b, c) - C(a, d) + C(a, c) \geq 0$$

or equivalently when C is continuous, the copula density function $c(u, v)$ is non-negative.

$$c(u, v) = \frac{\partial}{\partial u}\frac{\partial}{\partial v}C(u, v) \geq 0$$

3. Uniform Marginals: If all except 1 argument is 1, then the left-out argument becomes the value of C.

$$C(u, 1) = u, u \in [0, 1]$$

$$C(1, v) = v, v \in [0, 1]$$

Note: The definition above is extendable to more than 2 random variables and may be found in McNeil, et al (2015).

Some preliminaries before I move forward:

Quantile Function: For a random variable X with distribution function F_X, the quantile function $F_X^{\leftarrow} : [0, 1] \to \mathbb{R}$ such that:

$$F_X^{\leftarrow}(u) = \inf\{x : F_X(x) \geq u\}$$

Quantile Transform: For $U \sim U(0, 1)$ and distribution function F. If $X = F^{\leftarrow}(U)$, then X will have the distribution function F, i.e., $F_X(x) = F(x)$.

Probability Integral Transform: For $X \sim F_X$. If $U = F_X(X)$, then $U \sim U(0, 1)$

Sklar's Theorem

Given the preliminaries, we express the (bivariate) Sklar's theorem which is pivotal with how we can use copulas for data analysis and statistical modeling:

For random variables X and Y with joint distribution $F_{X,Y}$, there exists a copula C such that

$$F_{X,Y}(x, y) = C(F_X(x), F_Y(y))$$

Invariance Property of Copulas

Suppose X and Y are continuous random variables and their copula is defined as C. For continuous strictly increasing functions T_1 and T_2, the joint distribution of $T_1(X)$ and $T_2(Y)$ will also be C.

Frechet-Hoeffding Bounds

For a multivariate d-dimensional copula $C(u_1, \ldots, u_d)$,

$$\max\left\{\left[\sum_{j=1}^{d} u_j\right] - d + 1, 0\right\} \leq C(u_1, \ldots, u_d) \leq \min\{u_1, \ldots, u_d\}$$

Comment:

1. For $d = 2$, the left bound is a special copula called *countermonotone copula*, $C(u, v) = \max\{u + v - 1, 0\}$. It is indicative of a *perfect negative dependence* between two variables. It is not a copula when $d > 2$.
2. The right bound is known as the *comonotone copula* which indicates *perfect positive dependence* between variables.

Exchangeability

In some copula structures, it can be described that a set of random variables are not necessarily independent but they are *exchangeable* when:

$$(X_1, \ldots, X_d)' =^{dist} (X_{\Pi(1)}, \ldots, X_{\Pi(d)})'$$

where $(\Pi(1), \ldots, \Pi(d))$ are permutations of numbers $1, \ldots, d$

Different Copulas

Fundamental Copulas

The countermonotone and comonotone copulas are known as fundamental copulas (McNeil 2015), since are they are implied by the basic theories of probability.

Another is the *independence copula*, defined as:

$$C(u_1, \ldots, u_d) = \prod_{i=1}^{d} u_i = u_1 \times \ldots \times u_d$$

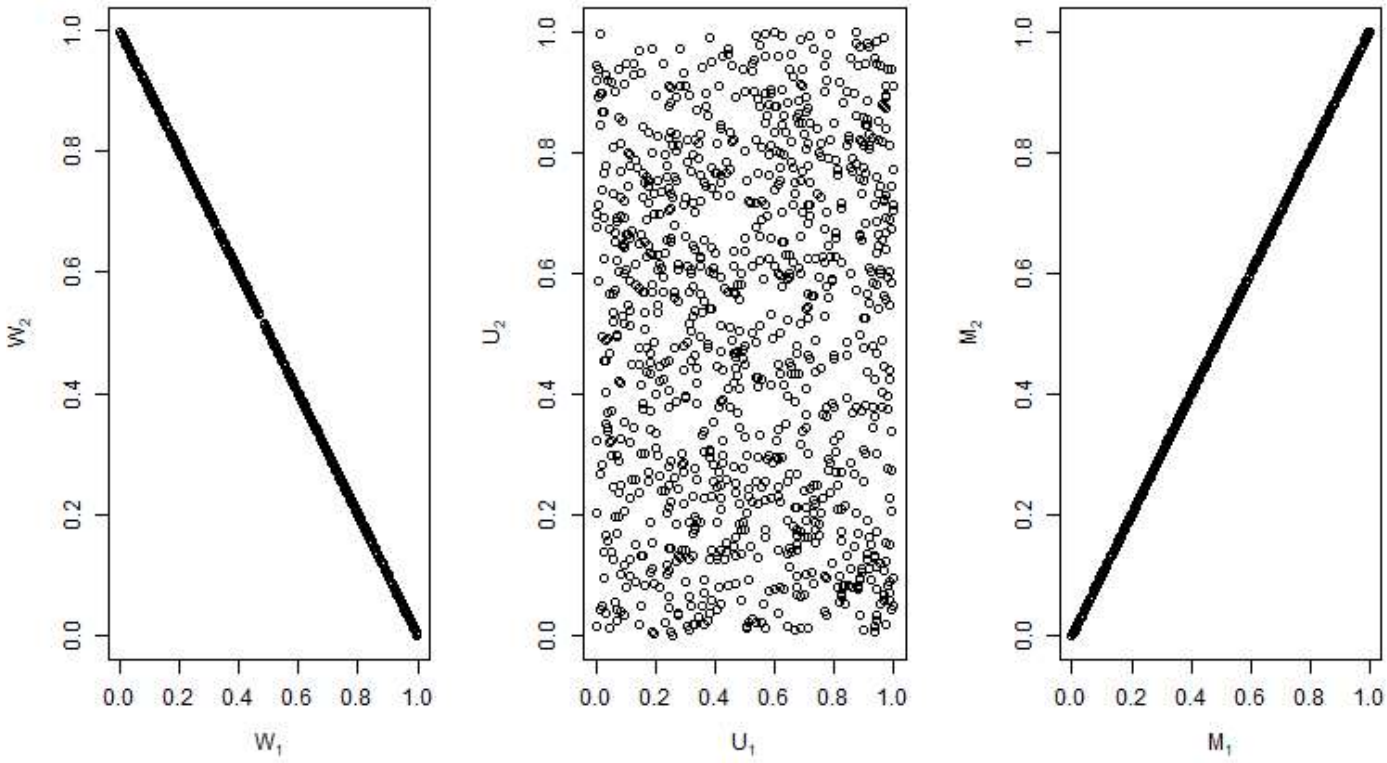
To show the three fundamental copulas, we have the following code:

Hide

```
n <- 1000 ## Sample size
set.seed(1954) ## Seed for reproducibility
W <- rCopula(n, lowfhCopula(2)) ## Generate the countermonotone copula
U <- rCopula(n, indepCopula(2)) ## Generate a sample from the independence copula
M <- rCopula(n, upfhCopula(2)) ## Generate the comonotone copula

par(mfrow =c(1,3))
## Plots of the Data
plot(W, xlab = quote(W[1]), ylab = quote(W[2]))
plot(U, xlab = quote(U[1]), ylab = quote(U[2]))
```

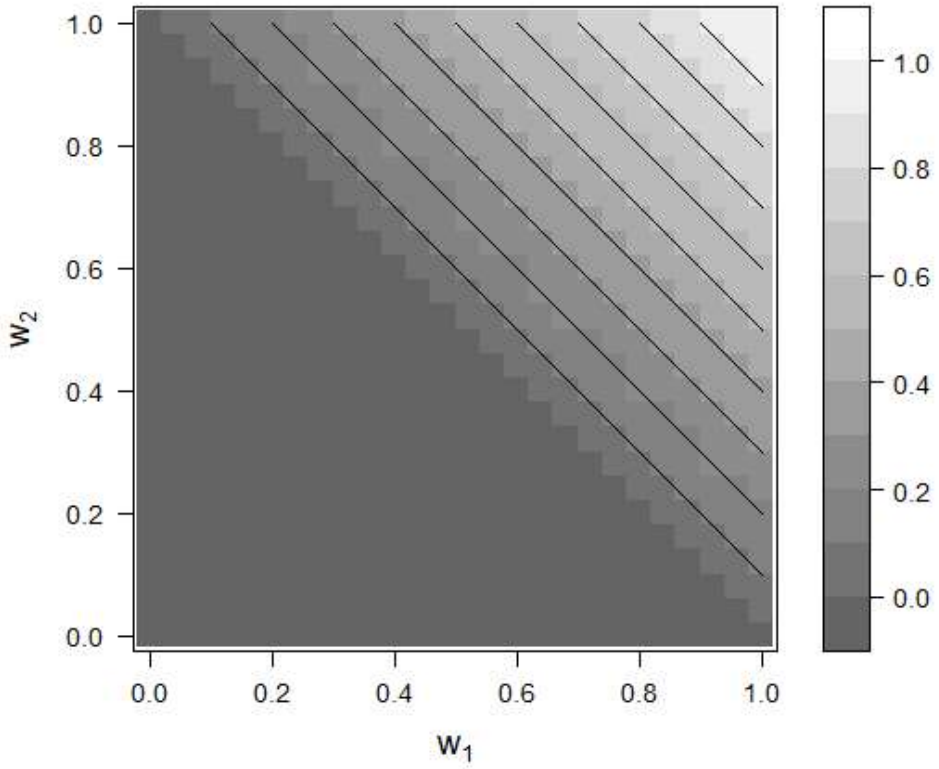
Hide

```
plot(M, xlab = quote(M[1]), ylab = quote(M[2]))
```



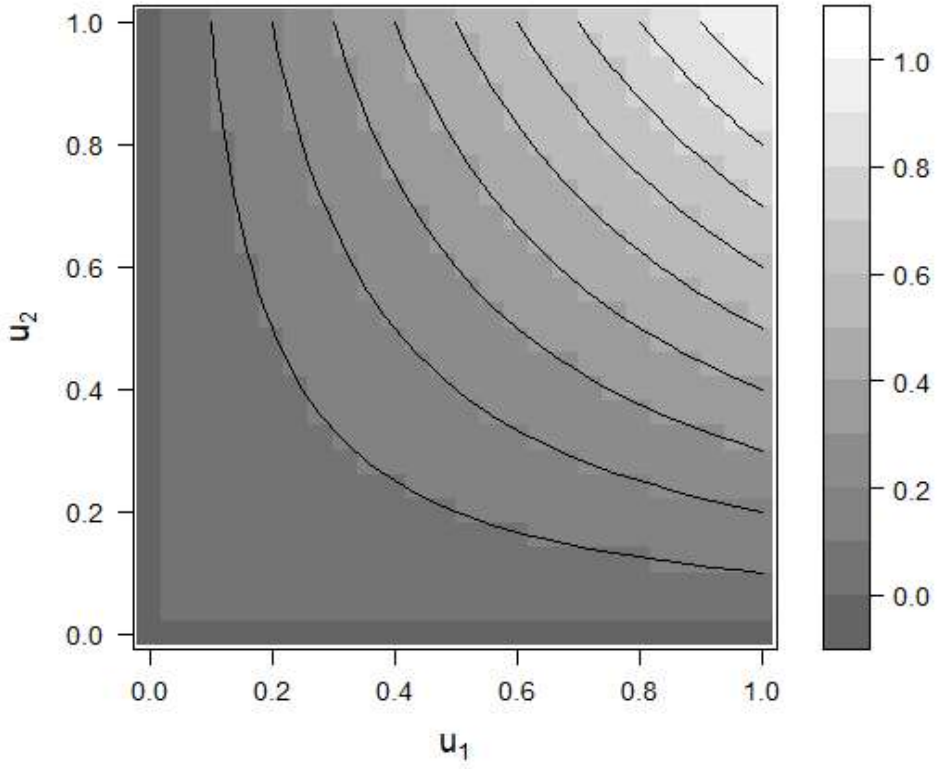
Hide

```
## Contour Plots for Copulas
contourplot2(lowfhCopula(2), pCopula, xlab = quote(w[1]),  ylab = quote(w[2]))
```



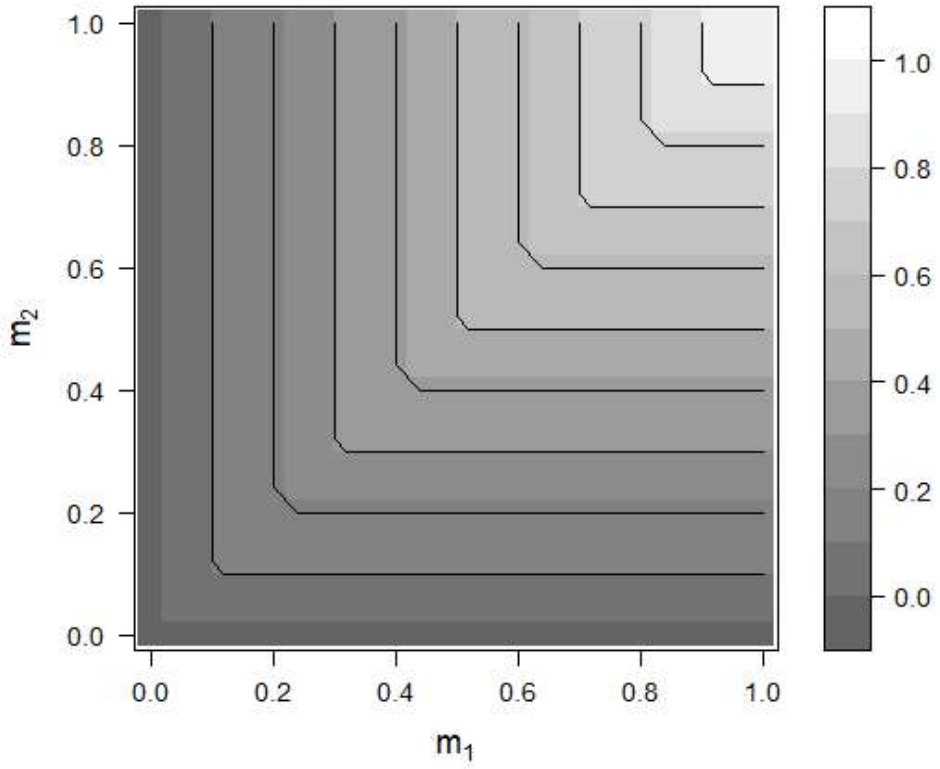
Hide

```
contourplot2(indepCopula(2), pCopula)
```



Hide

```
contourplot2(upfhCopula(2), pCopula, xlab = quote(m[1]),  ylab = quote(m[2]))
```



Implicit Copulas

Implicit copulas are copulas based on known multivariate distributions.

1. Gaussian copula: based on the multivariate Gaussian (normal) distribution. let $X \sim N_d(\mu = \mathbf{0}, \mathbf{\Sigma} = \mathbf{P})$, where P is a correlation matrix and $\mathbf{\Phi_P}(x_1, \ldots, x_d)$ is the joint distribution described solely by the correlation coefficient and Φ is the univariate standard normal distribution function. Then,

$$C_{\mathbf{P}}^{Ga}(u_1, \ldots u_d|\mathbf{P}) = \mathbf{\Phi_P}(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_d))$$

Note:

a. The Gaussian copula density function is shown below, where $\mathbf{x} = (\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_d))'$:

$$c_{\mathbf{P}}^{Ga} = \frac{1}{\sqrt{|\mathbf{P}|}} \exp\left(-\frac{1}{2}\mathbf{x}'(\mathbf{P}^{-1} - \mathbf{I})\mathbf{x}\right)$$

b. If $\mathbf{P} = \mathbf{I}$, then it becomes the independence copula.

2. t copula: based on the multvariate t distribution.let $X \sim t_d(\nu, \mu = \mathbf{0}, \mathbf{\Sigma} = \mathbf{P})$, where P is a correlation matrix and $\mathbf{t}_{\nu,\mathbf{P}}(x_1, \ldots, x_d)$ is the joint distribution described solely by the correlation coefficient and t_ν is the univariate t distribution function. Then,

$$C_{\nu,\mathbf{P}}^{t}(u_1, \ldots u_d|\mathbf{P}) = \mathbf{t}_{\nu,\mathbf{P}}(t_\nu^{-1}(u_1), \ldots, t_\nu^{-1}(u_d))$$

Note:

1. For both Gaussian and t copulas,

a. If $\mathbf{P} = \mathbf{1}\mathbf{1}'$, then they become the comonotone copula.

b. If $d = 2$ and $\rho_{12} = -1$, then they reduce to the countermonotone copula.

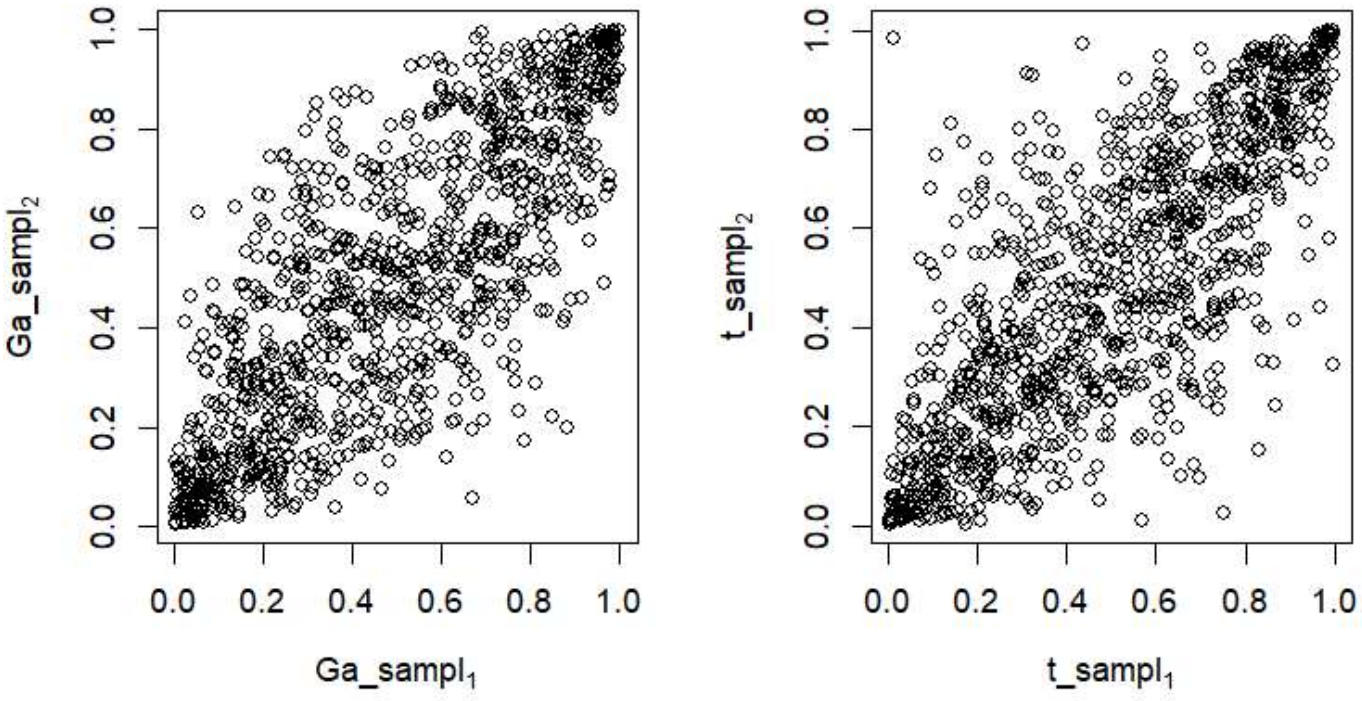
2. The Gaussian and t copulas are a special case of what are called *elliptical copulas* from the family of elliptical distributions.

3. From the implicit structure of Gaussian mixture models, one can also make asymmetric copulas such as the skewed normal and the skewed t copulas, but they are outside the scope of this introduction.

The code below visualizes the Gaussian and t copulas.

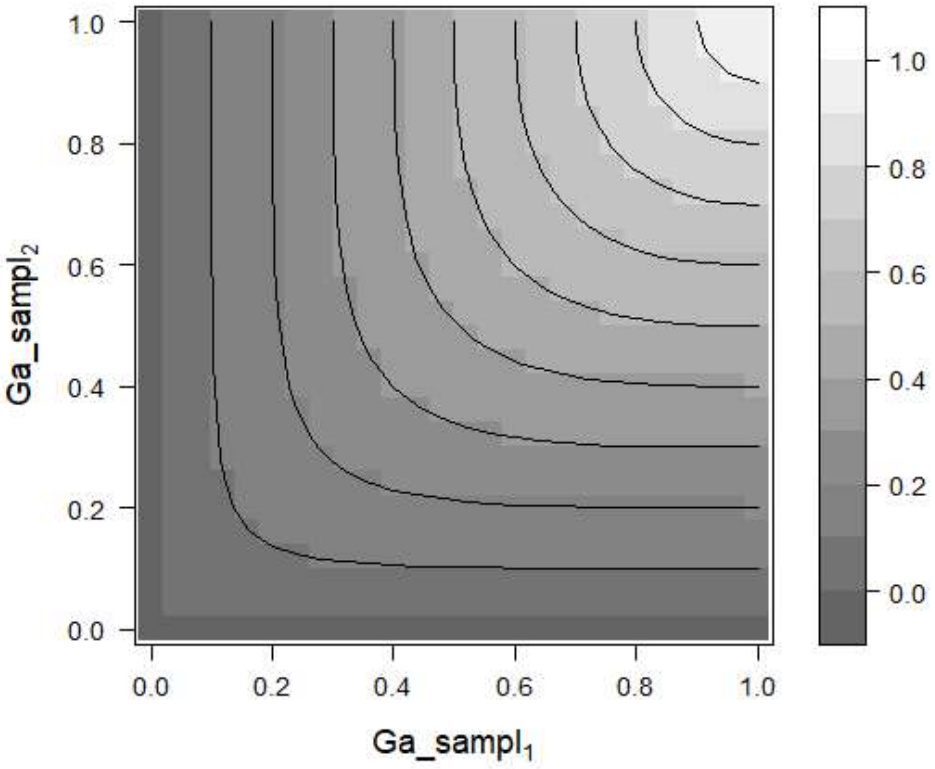
Hide

```
n <- 1000 ## Sample size
rho <- 0.8
set.seed(1954) ## Seed for reproducibility
Ga_sampl <- rCopula(n, normalCopula(rho , dim = 2)) ## Generate a sample from the Gaussian copula
t_sampl <- rCopula(n, tCopula(rho , dim = 2)) ## Generate a sample from the t copula

par(mfrow =c(1,2))
## Plots of the Data
plot(Ga_sampl, xlab = quote(Ga_sampl[1]), ylab = quote(Ga_sampl[2]))
plot(t_sampl, xlab = quote(t_sampl[1]), ylab = quote(t_sampl[2]))
```



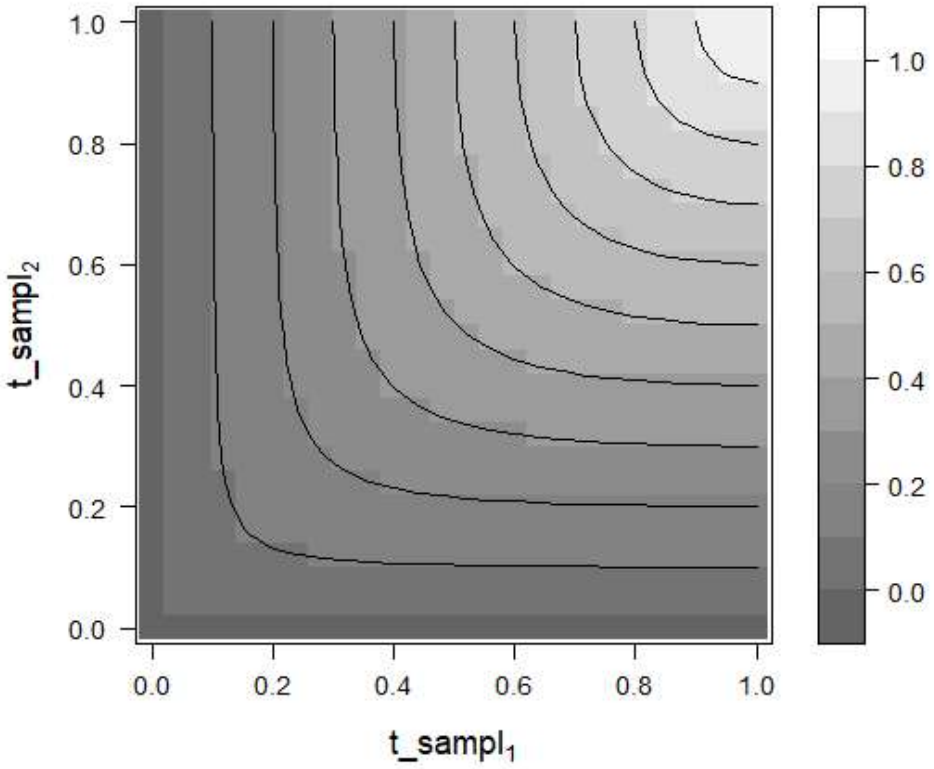
Hide

```
## Contour Plots for Copulas
contourplot2(normalCopula(rho , dim = 2), pCopula,  xlab = quote(Ga_sampl[1]), ylab = quote(Ga_sampl
[2]))
```



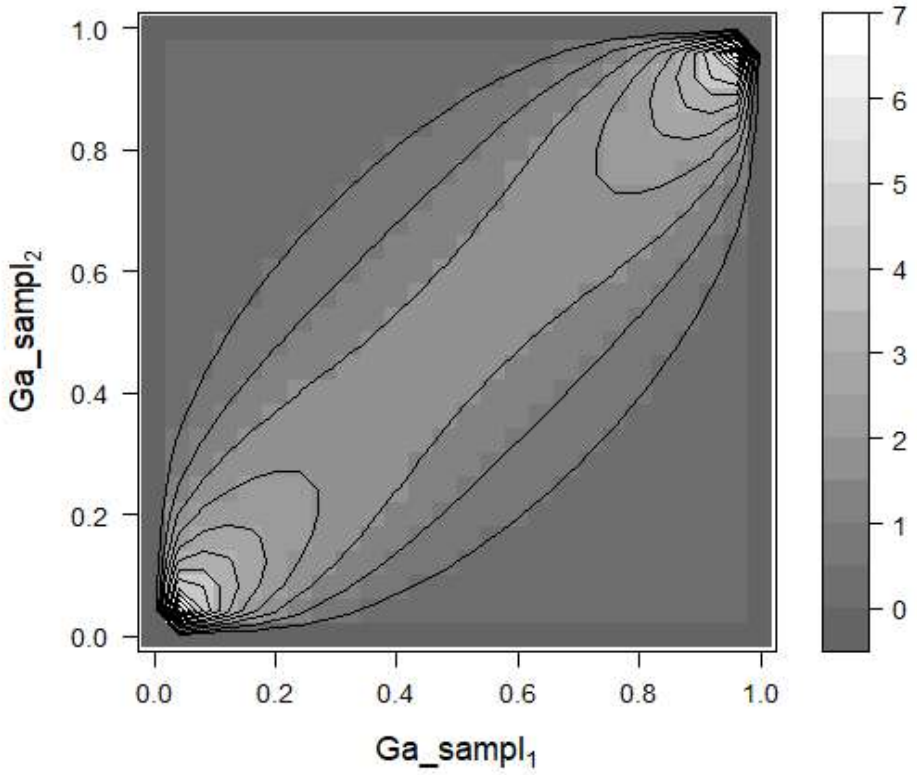
Hide

```
contourplot2( tCopula(rho , dim = 2), pCopula,  xlab = quote(t_sampl[1]), ylab = quote(t_sampl[2]))
```



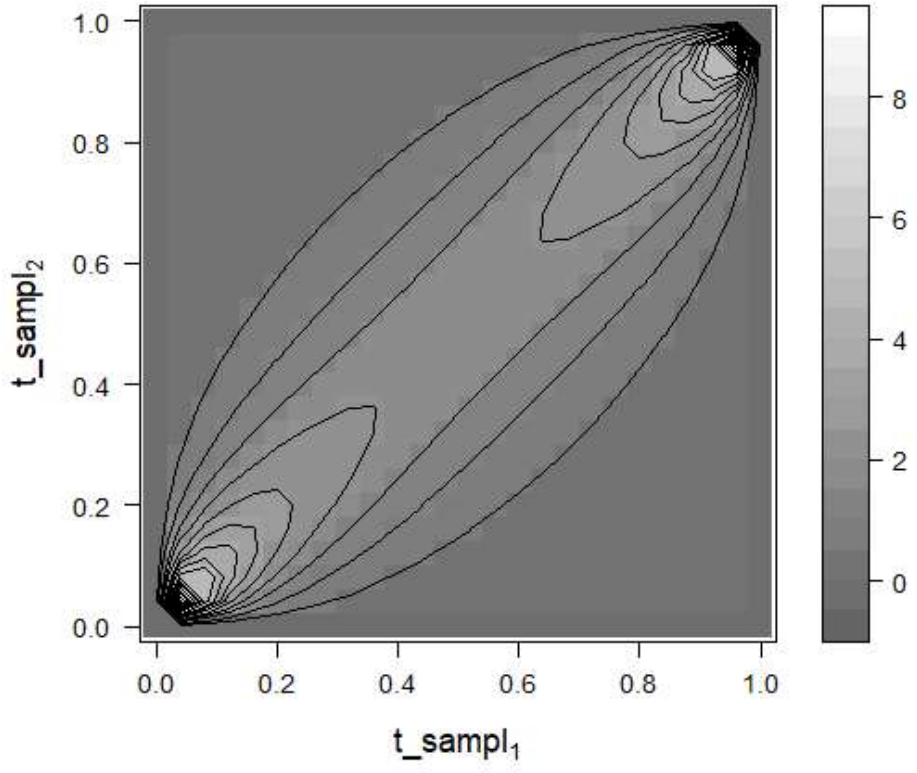
Hide

```
## Contour Plot for Copula Densities
contourplot2(normalCopula(rho , dim = 2), dCopula,  xlab = quote(Ga_sampl[1]), ylab = quote(Ga_sampl
[2]))
```



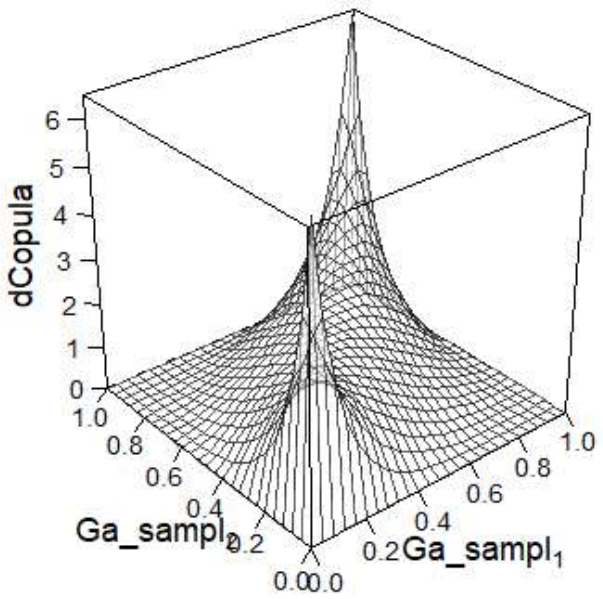
Hide

```
contourplot2( tCopula(rho , dim = 2), dCopula,  xlab = quote(t_sampl[1]), ylab = quote(t_sampl[2]))
```



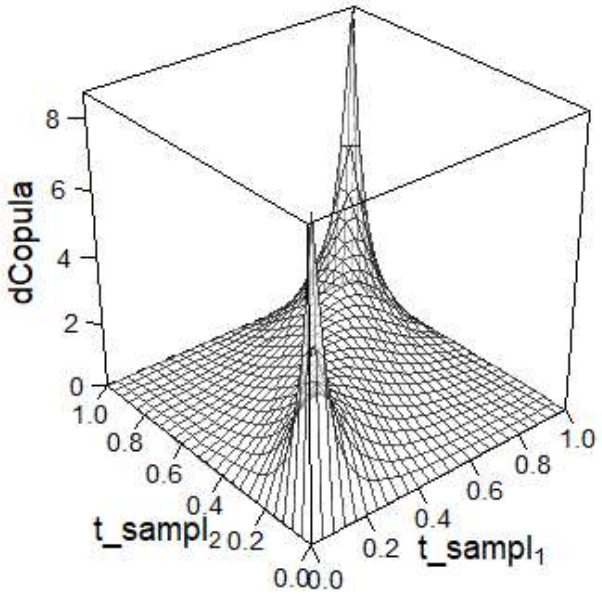
Hide

```
## Wireframe Plot for Copula Densities
wireframe2(normalCopula(rho , dim = 2), dCopula,  xlab = quote(Ga_sampl[1]), ylab = quote(Ga_sampl
[2]))
```



Hide

```
wireframe2( tCopula(rho , dim = 2), dCopula,  xlab = quote(t_sampl[1]), ylab = quote(t_sampl[2]))
```



Explicit Copulas

Explicit copulas are copulas that are explicitly constructed for their purpose.

Archimedean copulas

Archimedean copulas are a special family of copulas in which they are constructed by a convex monotone function $\psi : [0, \infty) \to [0, 1]$ such that $\psi(0) = 1$ and $\psi(t) \to 0$ as $t \to \infty$:

$$C^{\psi}(u_1, u_2) = \psi\left(\psi^{-1}(u_1) + \psi^{-1}(u_2)\right)$$

They can be generalized to a multivariate form, though the associations are described only by 1 parameter of the generator function ψ:

$$C^{\psi}(u_1, \ldots, u_d) = \psi\left(\psi^{-1}(u_1) + \ldots + \psi^{-1}(u_d)\right)$$

From these, the following copulas can be generated:

1. Independence Copula

$$\psi^{\Pi}(t) = e^{-t}$$

2. Gumbel Copula

$$\psi_{\theta}^{Gu}(t) = \exp\left(-t^{1/\theta}\right) \quad , \quad \theta \geq 1$$

3. Clayton Copula

$$\psi_{\theta}^{Cl}(t) = \max\left((1 + \theta t)^{-1/\theta}, 0\right) \quad , \quad \theta \geq -1$$

4. Frank Copula

$$\psi_\theta^{Fr}(t) = -\frac{1}{\theta}\ln\left(1 + (e^{-\theta} - 1)e^{-t}\right) \quad , \quad \theta \in \mathbb{R}$$

5. Generalized Clayton copula

$$\psi_{\theta,\delta}^{GCl}(t) = \left(1 + \theta t^{1/\delta}\right)^{-1/\theta} \quad , \quad \theta \geq 0, \delta \geq 1$$

Note:

1. The Clayton and Frank copulas are known as *comprehensive copulas*, as they can span from the countermonotone (when $d = 2$) to the comonotone copula as their special cases.
2. Numerous copulas can be made so long as their generator functions comply with the required limit properties and convexity.

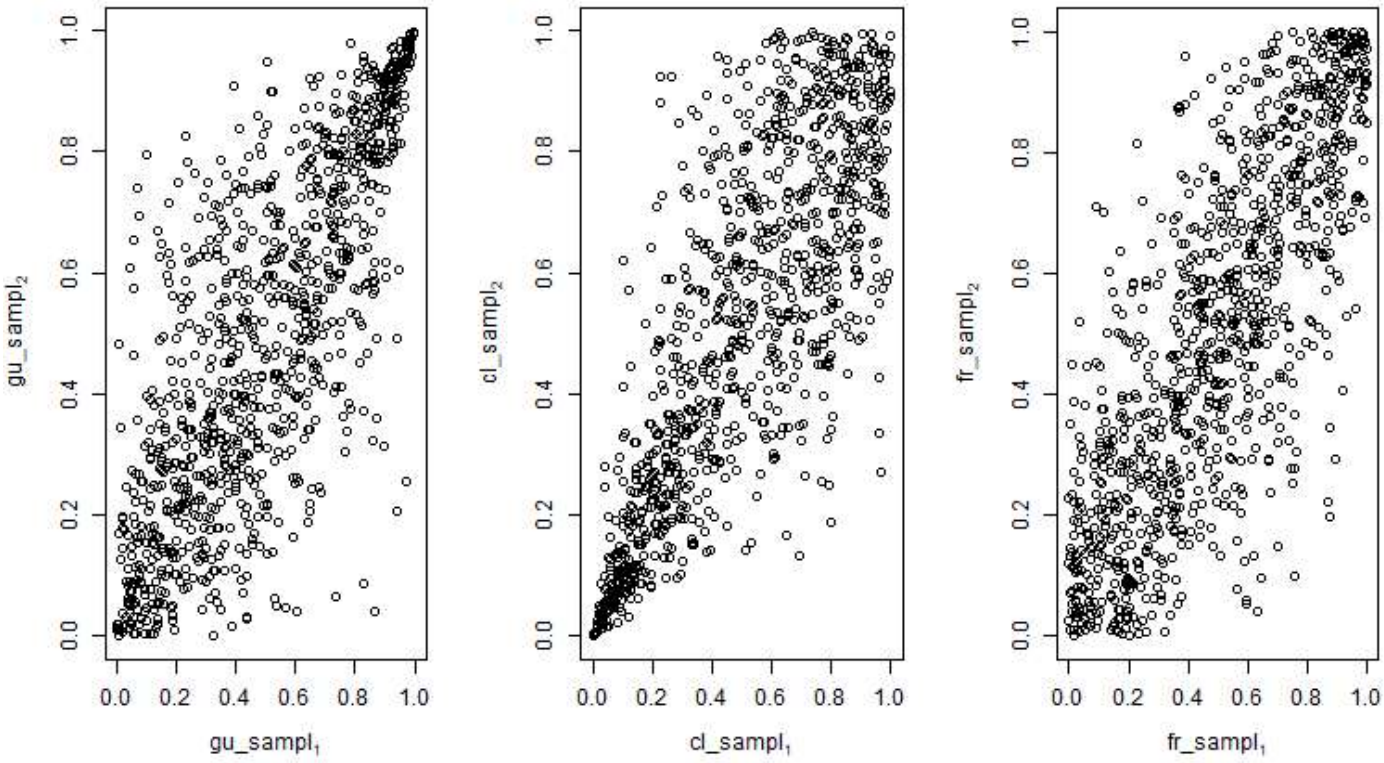
To visualize the Gumbel, Clayton, and Frank copulas, we have the code below:

Hide

```
tau_par<- 0.6 ## common Kendall's Tau value used for all copulas

## Aligning Theta for the similar tau
theta_gu <- iTau(gumbelCopula(), tau = tau_par)
theta_cl <- iTau(claytonCopula(), tau = tau_par)
theta_fr <- iTau(frankCopula(), tau = tau_par)

## generated copulas
c_gu <- gumbelCopula(theta_gu, dim = 2)
c_cl <- claytonCopula(theta_cl, dim = 2)
c_fr <- frankCopula(theta_fr, dim = 2)

## sample values
n <- 1000 ## Sample size

set.seed(1954) ## Seed for reproducibility

gu_sampl <- rCopula(n, c_gu) ## Generate a sample from the Gumbel copula
cl_sampl <- rCopula(n, c_cl) ## Generate a sample from the Clayton copula
fr_sampl <- rCopula(n, c_fr) ## Generate a sample from the Clayton copula

par(mfrow =c(1,3))
## Plots of the Data
plot(gu_sampl, xlab = quote(gu_sampl[1]), ylab = quote(gu_sampl[2]))
plot(cl_sampl, xlab = quote(cl_sampl[1]), ylab = quote(cl_sampl[2]))
```

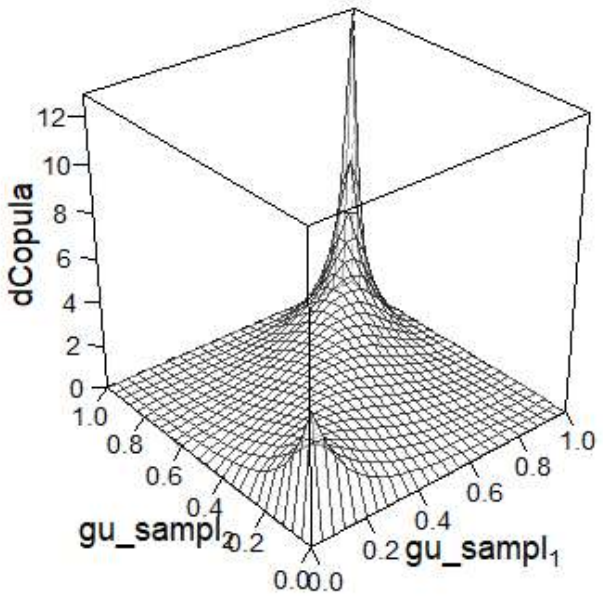
Hide

```
plot(fr_sampl, xlab = quote(fr_sampl[1]), ylab = quote(fr_sampl[2]))
```



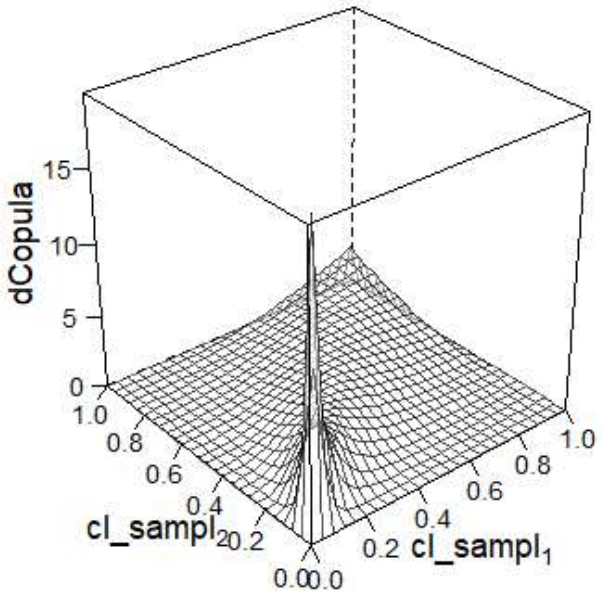
Hide

```
## wireframe density plots
wireframe2(c_gu, FUN = dCopula, xlab = quote(gu_sampl[1]), ylab = quote(gu_sampl[2]))
```



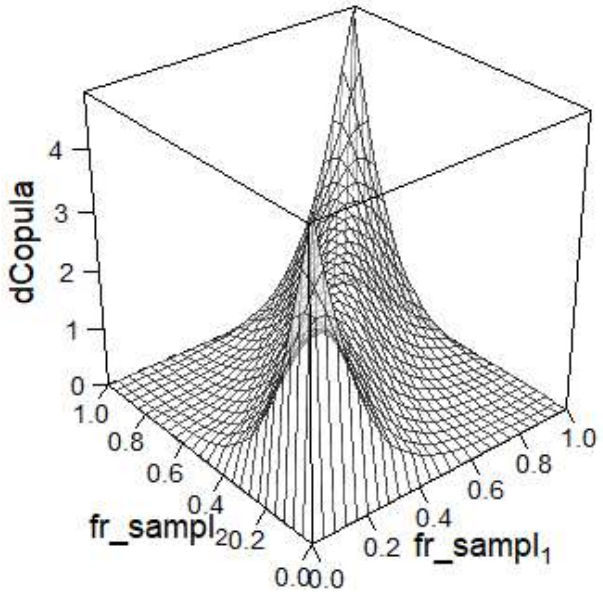
Hide

```
wireframe2(c_cl, FUN = dCopula, xlab = quote(cl_sampl[1]), ylab = quote(cl_sampl[2]))
```



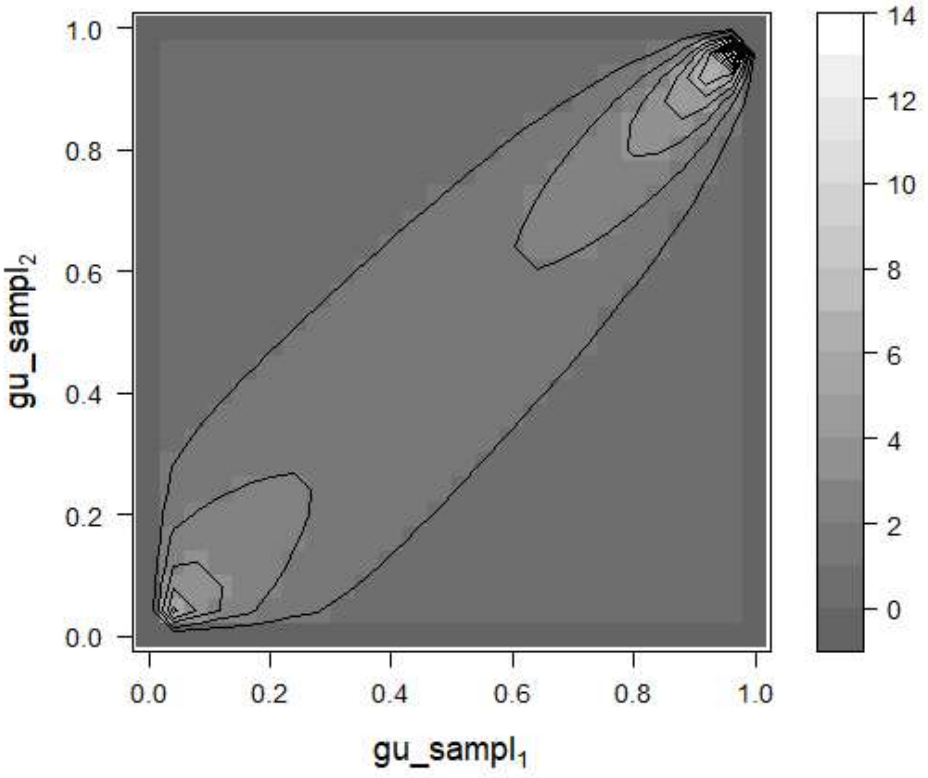
Hide

```
wireframe2(c_fr, FUN = dCopula, xlab = quote(fr_sampl[1]), ylab = quote(fr_sampl[2]))
```



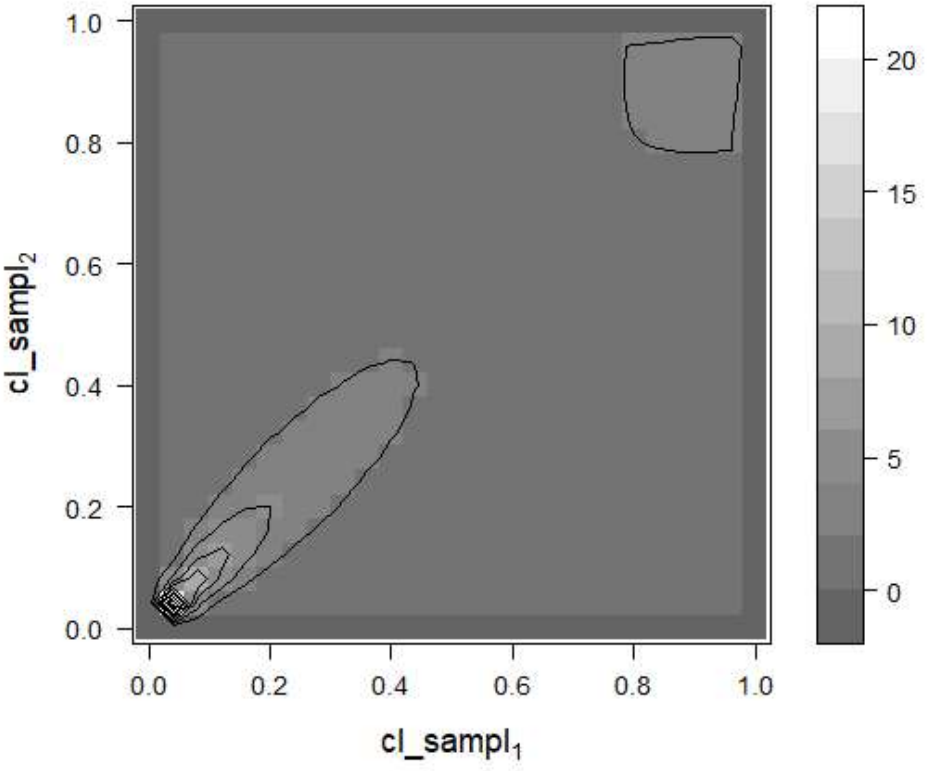
Hide

```
## contour density plots
contourplot2(c_gu, FUN = dCopula, xlab = quote(gu_sampl[1]), ylab = quote(gu_sampl[2]))
```



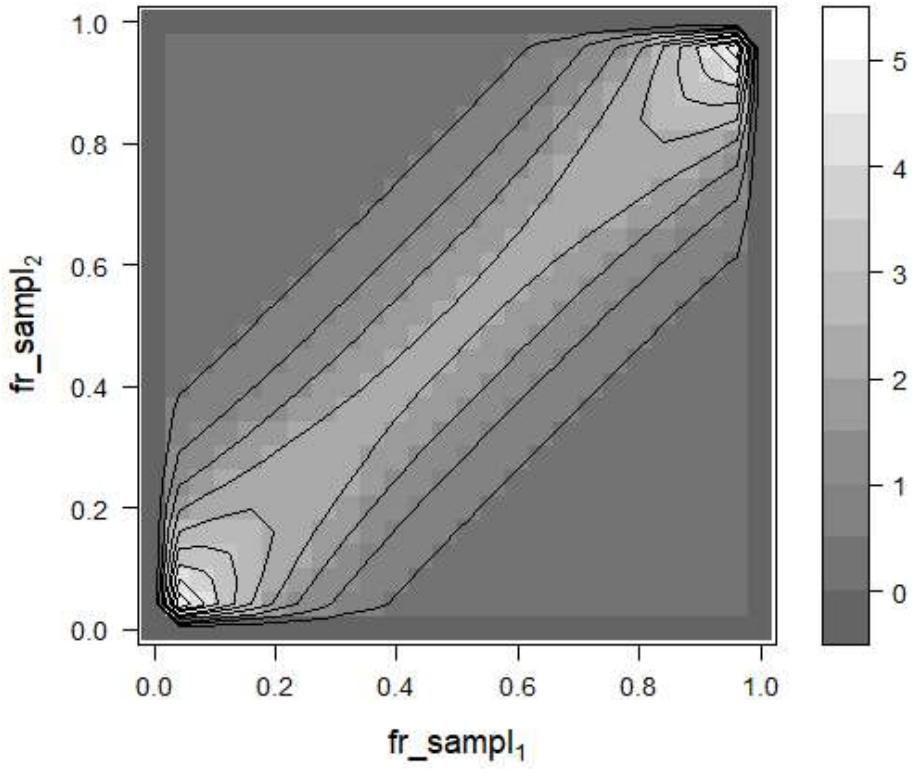
Hide

```
contourplot2(c_cl, FUN = dCopula, xlab = quote(cl_sampl[1]), ylab = quote(cl_sampl[2]))
```



Hide

```
contourplot2(c_fr, FUN = dCopula, xlab = quote(fr_sampl[1]), ylab = quote(fr_sampl[2]))
```



Survival Copulas

Survival copulas are analogous to survival distributions. We will only talk about bivariate survival copulas as they are easily derived.

A survival copula $\hat{C}$ for X and Y is defined as:

$$P(X > x, Y > y) = \hat{C}(1 - F_X(x), 1 - F_Y(y))$$

For a bivariate copula C, its survival copula $\hat{C}$ can be derived from the equality:

$$\hat{C}(1 - u_1, 1 - u_2) = 1 - u_1 - u_2 + C(u_1, u_2)$$

They are commonly in finance and environmental statistics when studying maximum values from 2 or more data variables.

Association Measures Based on Copulas

Kendall's tau

Denoted by ρ_τ, the Kendall's tau between X_1 and X_2 is

$$\rho_\tau(X_1, X_2) = E[sign((X_1 - \mu_1)(X_1 - \mu_2))]$$

Kendall's tau is often described as a measure of *concordance* or agreement.

The copulas can be used to derive Kendall's tau, specifically:

$$\rho_\tau(X_1, X_2) = \left[4 \int_0^1 \int_0^1 C(u,v) dC(u,v)\right] - 1$$

Spearman's rank correlation

The nonparametric analogue of Pearson correlation, called Spearman's correlation ρ_S, is simply defined as

$$\rho_S(X_1, X_2) = \rho(F_{X_1}(X_1), F_{X_2}(X_2))$$

In terms of copulas, then

$$\rho_S(X_1, X_2) = \left[12 \int_0^1 \int_0^1 C(u,v) du dv\right] - 3$$

Tail Dependence

It is possible that the tails of the joint distribution have a different dependence description than around the means of the distribution. For example, the variables may be uncorrelated around the mean but extreme observations tend to happen at the same time. This is a common occurrence in financial crises and in extreme environment scenarios.

The phenomenon in which the tails of distribution have some dependence features is called *tail dependence*.

Two tail dependence measures are being looked: lower tail dependence λ_l and upper tail dependence λ_u

The definition of the measures are:

$$\lambda_l = \lim_{q \to 0^+} \frac{C(q,q)}{q}$$

$$\lambda_u = \lim_{q \to 0^+} \frac{\hat{C}(q,q)}{q}$$

When the tail dependence values are equal to zero, then it means that there is asymptotic tail independence in that side of the tail.

For some Archimedean copulas, their dependence measure values are given below:

Table 7.5. Kendall's rank correlations and coefficients of tail dependence for the copulas of Table 7.4. $D_1(\theta)$ is the Debye function $D_1(\theta) = \theta^{-1} \int_0^\theta t/(e^t - 1)\, dt$.

Copula	ρ_τ	λ_u	λ_l
C_θ^{Gu}	$1 - 1/\theta$	$2 - 2^{1/\theta}$	0
C_θ^{Cl}	$\theta/(\theta+2)$	0	$\begin{cases} 2^{-1/\theta}, & \theta > 0, \\ 0, & \theta \leqslant 0, \end{cases}$
C_θ^{Fr}	$1 - 4\theta^{-1}(1 - D_1(\theta))$	0	0
$C_{\theta,\delta}^{GC}$	$\frac{(2+\theta)\delta - 2}{(2+\theta)\delta}$	$2 - 2^{1/\delta}$	$2^{-1/(\theta\delta)}$

Dependence Values for Some Archimedean Copulas (McNeil 2015)

Estimating Copulas

Nonparametric Copula

We can already generate the nonparametric copula by transforming the data into *pseudo-observations*, which transform the data into uniform variables

For $(X_{i,1}, \ldots, X_{i,d}), i = 1, 2, \ldots, n,$

$$u_{i,j} = \frac{r_{i,j}}{n+1}$$

where $r_{i,j} =$ rank of the i^{th} observation in the j^{th} variable.

From the pseudo-observations, one can grasp some idea of the shape of the copula.

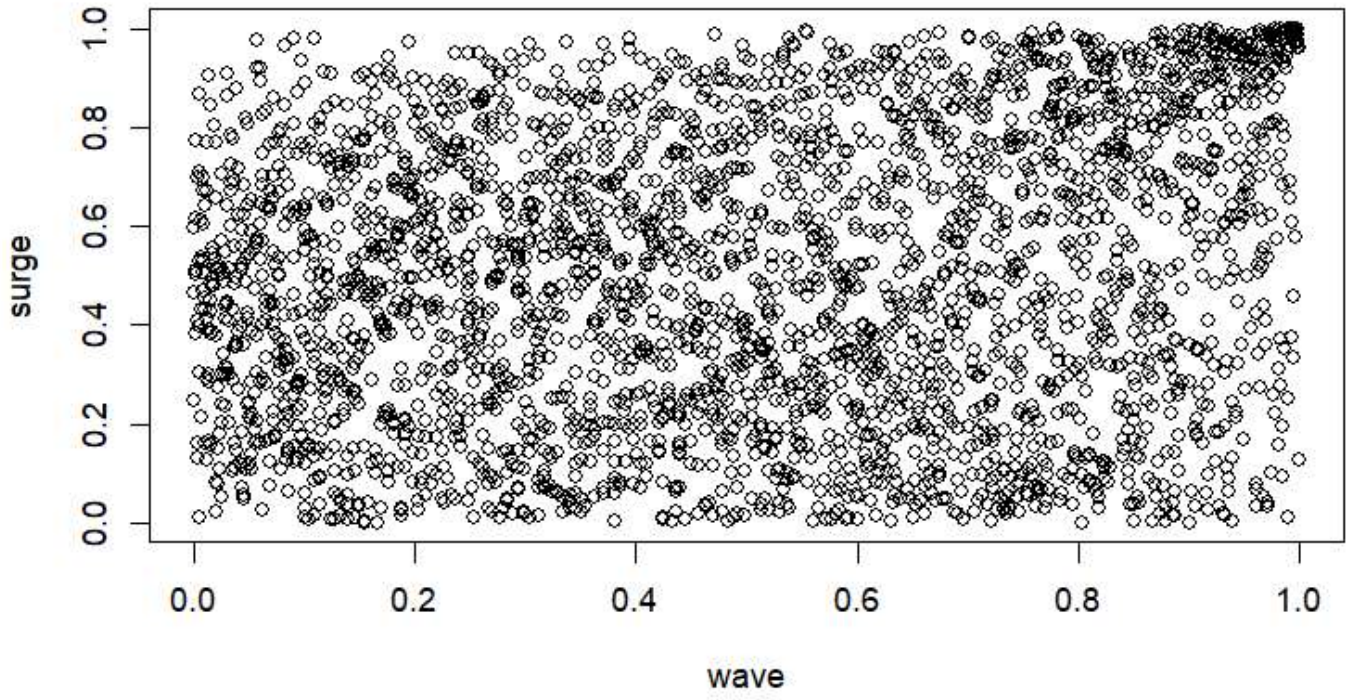
Hide

```
## install.packages("MASS")

library(MASS)
```

```
Attaching package: ‘MASS’

The following object is masked _by_ ‘.GlobalEnv’:

    SP500

The following object is masked from ‘package:qqtest’:

    bacteria

The following object is masked from ‘package:dplyr’:

    select

The following objects are masked from ‘package:fma’:

    cement, housing, petrol
```

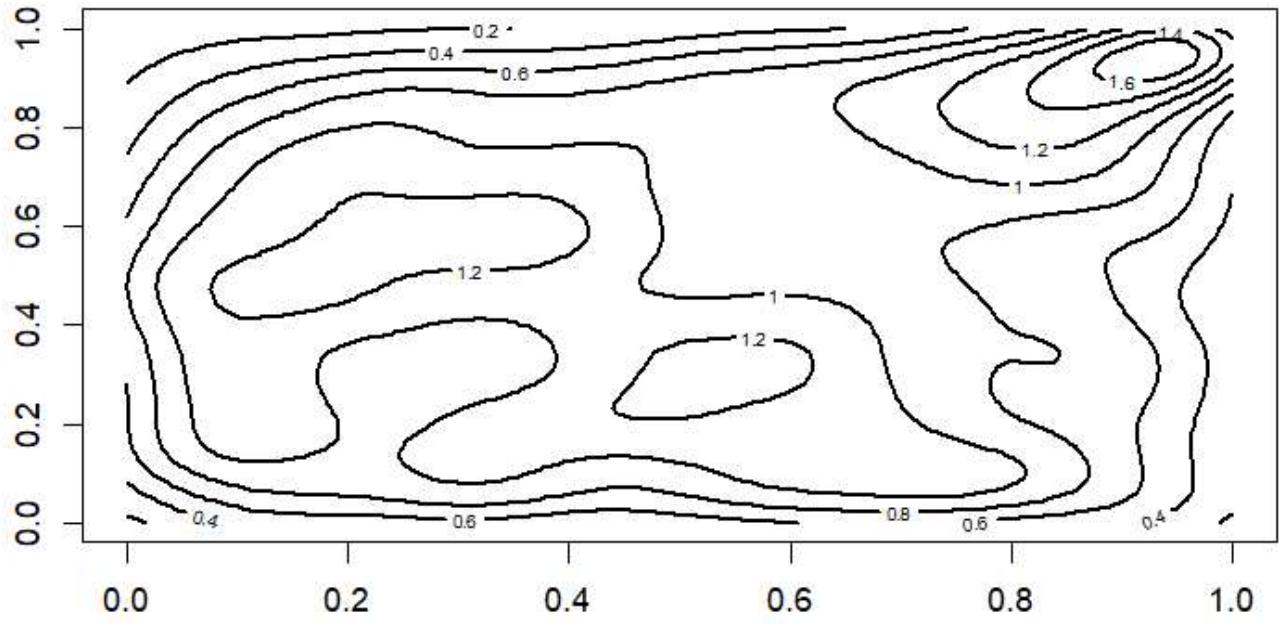
Hide

```
data(wavesurge)

pobs_ws <- pobs(wavesurge)

par(mfrow=c(1,1))
plot(pobs_ws)
```



Hide

```
z <- kde2d(pobs_ws[,1], pobs_ws[,2], n = 100)

## contour plot
contour(z, lwd =2)
```



Fitting Copulas

You can fit copulas via the following

1. Methods of Moments Matching With Association Measures: A candidate copula is fit by matching its known association measure with the association measure value of the data. For example, matching the Kendall's tau of the data with a known copula with parameter θ then solve for $theta$.

$$\hat{\rho}_\tau = \rho_\tau^{Copula}(\theta) \quad \Rightarrow \quad \tilde{\theta}_{MM} = \tilde{\theta}(\hat{\rho}_\tau)$$

Note: If there are more than 1 parameters in the assumed copula, one may use more association measures.

2. Maximum Likelihood Estimation

We can fit a specific copula by estimating its parameters using the pseudo-observations of the data and defining a log-likelihood function.

The estimator would be:

$$\hat{\theta}_{ML} = \arg\sup_\theta \ln L(\theta|\underline{u}_1, \ldots, \underline{u}_n)$$

with the likelihood function defined as

$$\ln L(\theta|\mathbf{u}_1, \ldots, \mathbf{u}_n) = \sum_{i=1}^{n} \ln c_\theta(\mathbf{u}_i)$$

Note:

There are other methods for estimating copulas; it is a budding field as well!

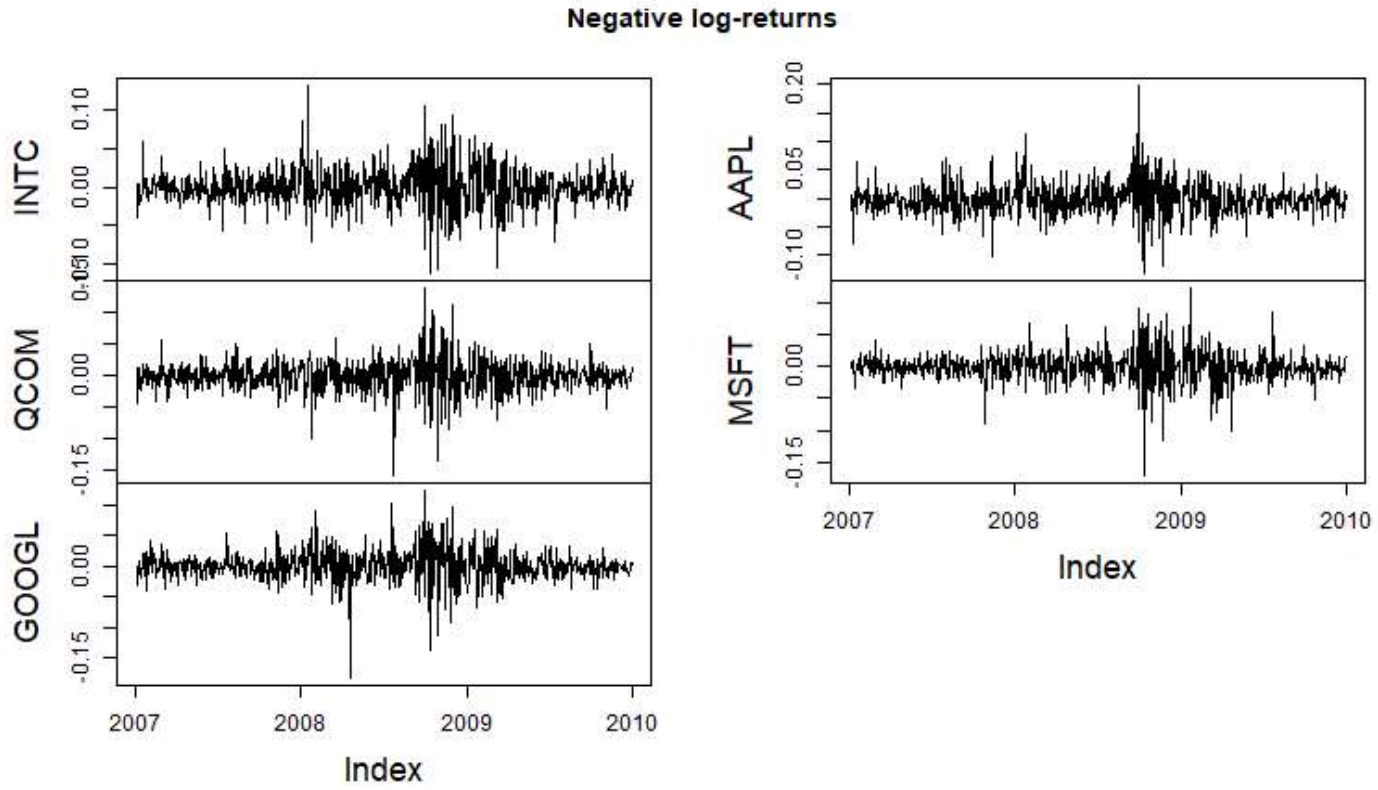
Example from Finance

In this example, we estimate the value-at-risk of a portfolio.

Reference:
https://github.com/qrmtutorial/qrm/blob/master/code/07_Copulas_and_Dependence/07_Copula_estimation_and_goodness-of-fit.R
(https://github.com/qrmtutorial/qrm/blob/master/code/07_Copulas_and_Dependence/07_Copula_estimation_and_goodness-of-fit.R)

Hide

```
### 1 Working with the data ####################################################

## Select the data we work with
data("SP500_const") # load the constituents data of the S&P 500
stocks <- c("INTC", "QCOM", "GOOGL", "AAPL", "MSFT") # Intel, Qualcomm, Google, Apple, Microsoft
time <- c("2007-01-03", "2009-12-31") # time period
S <- SP500_const[paste0(time, collapse = "/"), stocks] # data

## Check for missing data
stopifnot(all(!is.na(S))) # na.fill(, fill = "extend") is often helpful


### 2 Fitting marginal ARMA(1,1)-GARCH(1,1) models with standardized t residuals

## Build negative log-returns
X <- -returns(S) # -log-returns

## Basic plot
plot.zoo(X, main = "Negative log-returns")
```



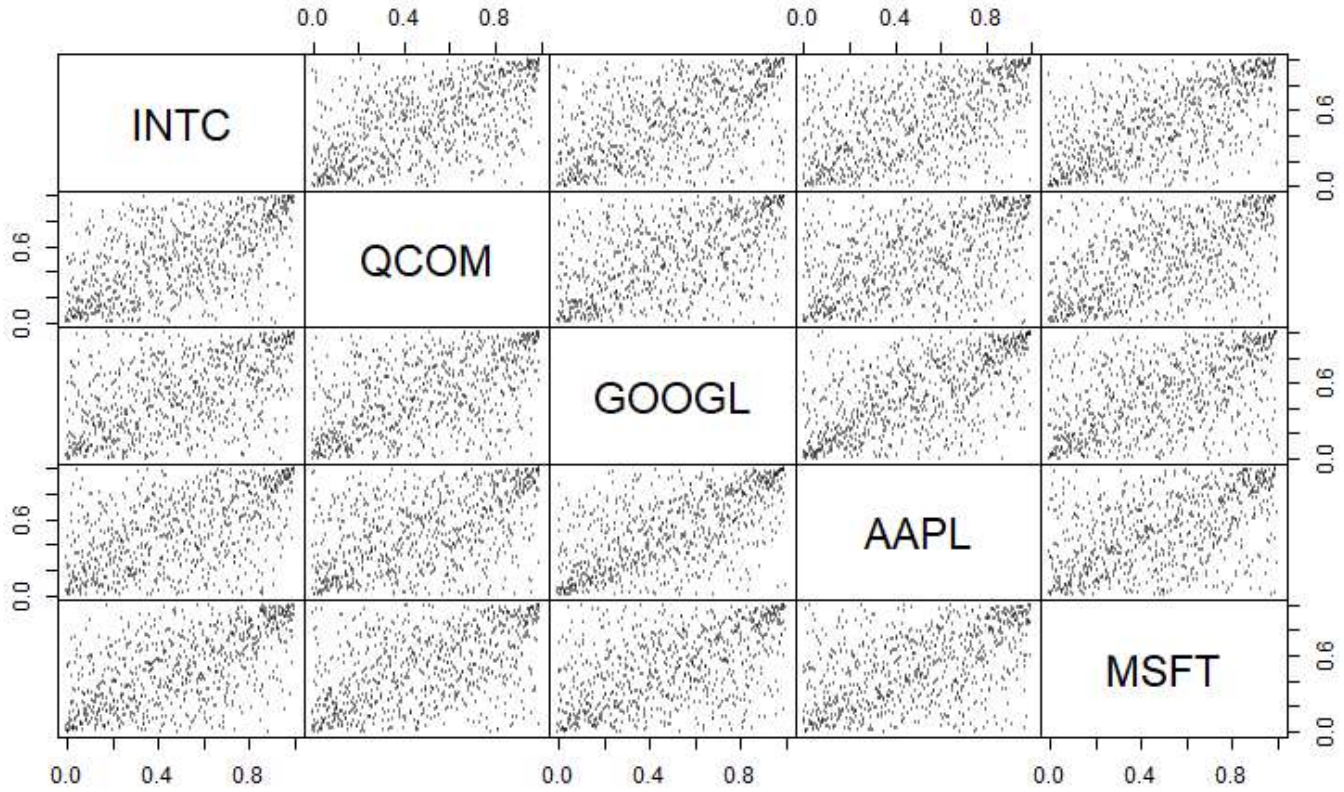
Hide

```
## Fit marginal time series
uspec <- rep(list(ugarchspec(distribution.model = "std")), ncol(X))
fit.ARMA.GARCH <- fit_ARMA_GARCH(X, ugarchspec.list = uspec)
stopifnot(sapply(fit.ARMA.GARCH$error, is.null)) # NULL = no error
if(FALSE)
    fit.ARMA.GARCH$warning
    ## => Warning comes from finding initial values and can be ignored here
fits <- fit.ARMA.GARCH$fit # fitted models
Z <- as.matrix(do.call(merge, lapply(fits, residuals, standardize = TRUE))) # grab out standardized re
siduals
colnames(Z) <- colnames(S)
(nu.mar <- vapply(fits, function(x) x@fit$coef[["shape"]], NA_real_)) # vector of estimated df
```

```
[1] 6.093997 7.150332 4.822862 7.507546 4.532550
```

Hide

```
n <- nrow(X) # sample size
d <- ncol(X) # dimension


### 3 Fitting copulas ###########################################################

## Compute pseudo-observations from the standardized t residuals
U <- pobs(Z)
pairs2(U, cex = 0.4, col = adjustcolor("black", alpha.f = 0.5))
```



Hide

```
## Fitting a Gumbel copula
fit.gc <- fitCopula(gumbelCopula(dim = d),
                    data = U, method = "mpl")
fit.gc@estimate # estimated copula parameter
```

```
[1] 1.454775
```

Hide

```
gc <- fit.gc@copula # fitted copula

## Compute matrices of pairwise Kendall's tau and upper tail-dependence coefficients
p2P(tau(gc), d = d)
```

```
          [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 1.0000000 0.3126087 0.3126087 0.3126087 0.3126087
[2,] 0.3126087 1.0000000 0.3126087 0.3126087 0.3126087
[3,] 0.3126087 0.3126087 1.0000000 0.3126087 0.3126087
[4,] 0.3126087 0.3126087 0.3126087 1.0000000 0.3126087
[5,] 0.3126087 0.3126087 0.3126087 0.3126087 1.0000000
```

Hide

```
p2P(lambda(gc)["upper"], d = d)
```

```
        [,1]     [,2]     [,3]     [,4]     [,5]
[1,] 1.000000 0.389631 0.389631 0.389631 0.389631
[2,] 0.389631 1.000000 0.389631 0.389631 0.389631
[3,] 0.389631 0.389631 1.000000 0.389631 0.389631
[4,] 0.389631 0.389631 0.389631 1.000000 0.389631
[5,] 0.389631 0.389631 0.389631 0.389631 1.000000
```

Hide

```
## Fitting a t copula
fit.tc <- fitCopula(tCopula(dim = d, dispstr = "un"),
                    data = U, method = "itau.mpl")
(nu <- tail(fit.tc@estimate, n = 1)) # estimated degrees of freedom nu
```

```
[1] 7.269531
```

Hide

```
(P <- p2P(head(fit.tc@estimate, n = -1))) # estimated correlation matrix
```

```
         [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 1.0000000 0.5761937 0.5246179 0.5361455 0.6246738
[2,] 0.5761937 1.0000000 0.5230008 0.5064793 0.5123869
[3,] 0.5246179 0.5230008 1.0000000 0.6429844 0.5391886
[4,] 0.5361455 0.5064793 0.6429844 1.0000000 0.5249092
[5,] 0.6246738 0.5123869 0.5391886 0.5249092 1.0000000
```

Hide

```
tc <- fit.tc@copula # fitted copula

## Compute matrices of pairwise Kendall's tau and upper tail-dependence coefficients
p2P(tau(tc))
```

```
         [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 1.0000000 0.3909252 0.3515836 0.3602403 0.4295361
[2,] 0.3909252 1.0000000 0.3503750 0.3381067 0.3424772
[3,] 0.3515836 0.3503750 1.0000000 0.4446080 0.3625380
[4,] 0.3602403 0.3381067 0.4446080 1.0000000 0.3518014
[5,] 0.4295361 0.3424772 0.3625380 0.3518014 1.0000000
```

Hide

```
p2P(lambda(tc)[(choose(d,2)+1):(d*(d-1))])
```

```
         [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 1.0000000 0.1730490 0.1457626 0.1514850 0.2031032
[2,] 0.1730490 1.0000000 0.1449762 0.1371633 0.1399111
[3,] 0.1457626 0.1449762 1.0000000 0.2157411 0.1530302
[4,] 0.1514850 0.1371633 0.2157411 1.0000000 0.1459047
[5,] 0.2031032 0.1399111 0.1530302 0.1459047 1.0000000
```

Hide

```
### 4 Goodness-of-fit ###########################################################

## We use the parametric bootstrap here, based on the maximum pseudo-likelihood
## estimator.
set.seed(271) # for reproducibility
N <- 100 # this is to save run time; it should be larger!

## Check the Gumbel copula
gof.gc <- gofCopula(gc, x = U, N = N) # warning also because the copula doesn't fit well here
```

```
================================
```

```
Warning: possible convergence problem: optim() gave code=52
```

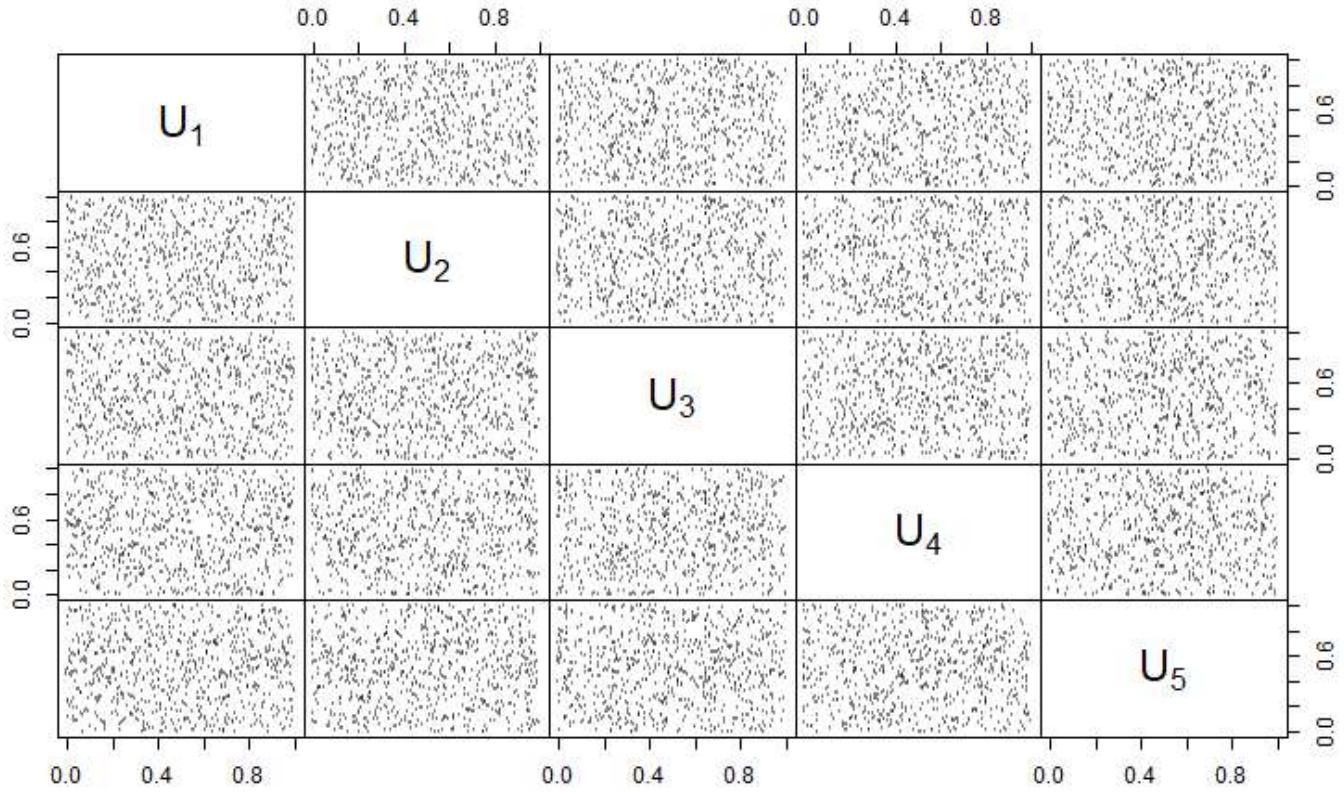
```
==================
```

```
Warning: possible convergence problem: optim() gave code=52
```

```
============================
```

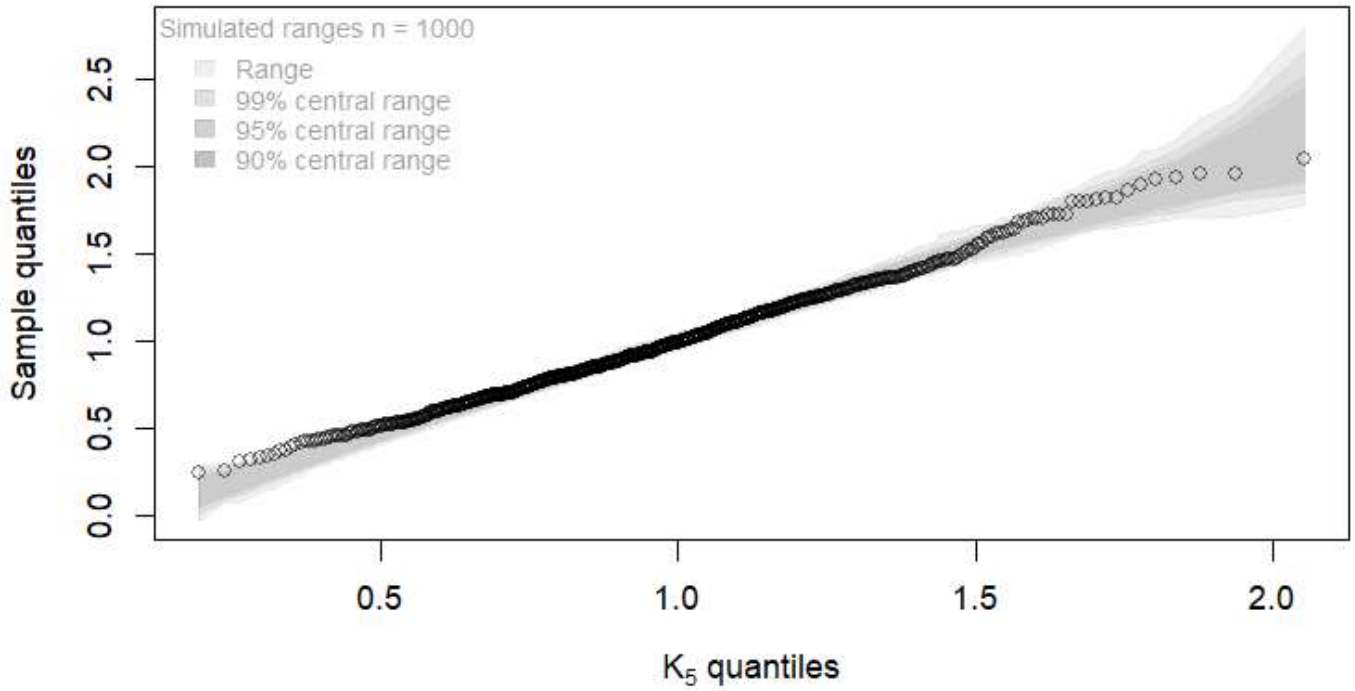
Hide

```
stopifnot(gof.gc$p.value < 0.05) # => rejection

## Check the t copula
## Note: - This can currently only be done for fixed and integer degrees of
##         freedom as there is no algorithm to evaluate the multivariate t df for
##         non-integer degrees of freedom.
##       - ... yet it's still quite slow. We thus check the model graphically
##         after mapping the variates to a U(0,1)^d distribution with the
##         Rosenblatt transform.
U.Rsnbl <- cCopula(U, copula = tc)
pairs2(U.Rsnbl, cex = 0.4, col = adjustcolor("black", alpha.f = 0.5)) # looks ok
```



Hide

```
## Map it to a K_d distribution ("kay") and do a Q-Q plot
U.Rsnbl.K <- sqrt(rowMeans(qnorm(U.Rsnbl)^2)) # map to a kay distribution
pK <- function(q, d) pchisq(d*q*q, df = d) # df of a K_d distribution
AD <- ad.test(U.Rsnbl.K, distr.fun = pK, d = d) # compute an AD test
stopifnot(AD$p.value >= 0.05)
## Note: The AD test here does not take into account parameter estimation
##       (that would require a parametric bootstrap, for example)

## A (sophisticated) Q-Q plot
qqtest(U.Rsnbl.K, dist = "kay", df = d, nreps = 1000, pch = 1,
       col = adjustcolor("black", alpha.f = 0.5), main = "",
       xlab = substitute(K[dof]~"quantiles", list(dof = d))) # => looks ok
```


Simulated ranges n = 1000
Range
99% central range
95% central range
90% central range
Sample quantiles
0.0
0.5
1.0
1.5
2.0
2.5
0.5
1.0
1.5
2.0
K_5 quantiles

Hide

```
### 5 Simulating paths from the full model ##################################

## Simulate paths
B <- 200
m <- ceiling(n/10) # length of the simulates paths
X.lst <- lapply(1:B, function(b) {
    ## 1) Simulate from the fitted copula
    U. <- rCopula(m, copula = tc)
    ## 2) Quantile-transform to standardized t distributions (for ugarchsim())
    Z. <- sapply(1:d, function(j) sqrt((nu.mar[j]-2)/nu.mar[j]) * qt(U.[,j], df = nu.mar[j]))
    ## 3) Use these multivariate dependent t innovations to sample from the
    ##    time series
    sim <- lapply(1:d, function(j)
        ugarchsim(fits[[j]], n.sim = m, m.sim = 1,
                  custom.dist = list(name = "sample",
                                     distfit = Z.[,j, drop = FALSE])))
    ## 4) Extract simulated series
    sapply(sim, function(x) fitted(x)) # simulated multivariate series X_t (= x@simulation$seriesSim)
})
## => List of length B containing (n x d)-matrices


### 6 Predict the aggregated loss and VaR_0.99 ##################################

## Note: - This is merely a demo of what can be done with the simulated data.
##       - See also the vignette "ARMA_GARCH_VaR" in qrmtools

## Predict VaR of the aggregated loss nonparametrically
Xs <- rowSums(X) # aggregated loss; n-vector
Xs. <- sapply(X.lst, rowSums) # simulated aggregated losses; (m, B)-matrix
Xs.mean <- rowMeans(Xs.) # predicted aggregated loss; m-vector
Xs.CI <- apply(Xs., 1, function(x) quantile(x, probs = c(0.025, 0.975))) # CIs; (2, m)-matrix
alpha <- 0.99 # confidence level
VaR <- apply(Xs., 1, function(x) quantile(x, probs = alpha)) # VaR_alpha; m-vector

## Plot
tm <- index(SP500_const)
start <- match(time[1], as.character(tm))
past <- tm[start:(start+n-1)]
future <- tm[(start+n):(start+n+m-1)]
plot(past, Xs, type = "l", xlim = range(c(past, future)), xlab = "", ylab = "") # actual (past) losses
polygon(c(future, rev(future)), c(Xs.CI[1,], rev(Xs.CI[2,])),
        border = NA, col = "grey80") # CI region
```

Hide

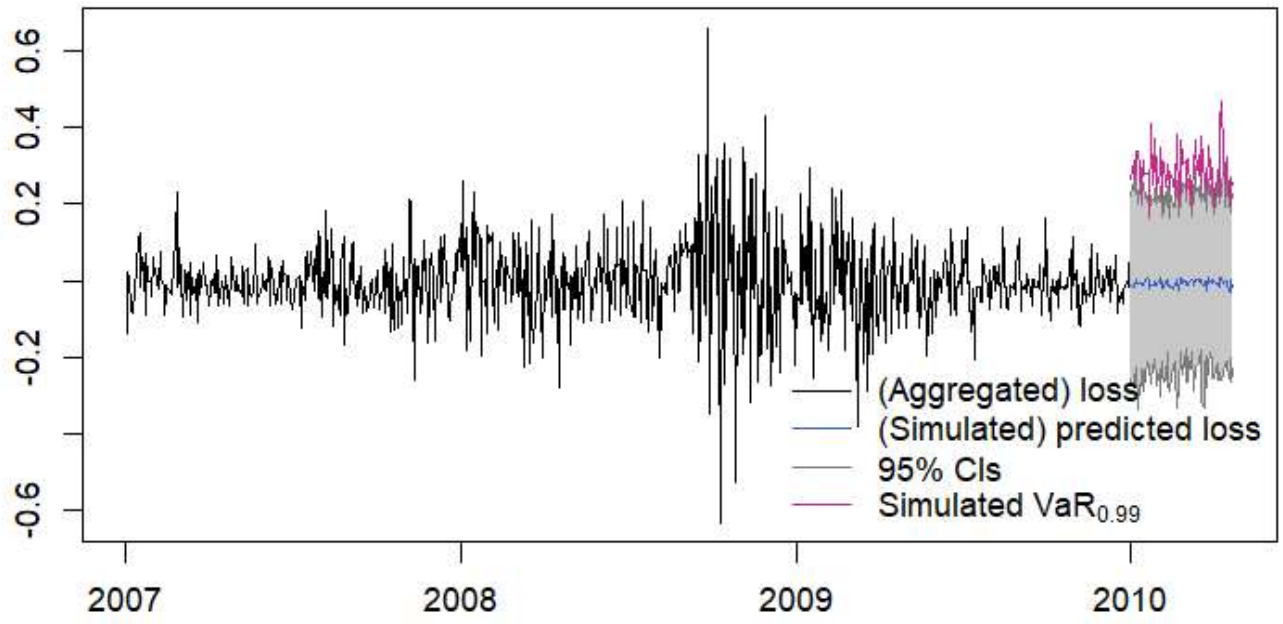
```
lines(future, Xs.mean, col = "royalblue3") # predicted aggregated loss
lines(future, Xs.CI[1,], col = "grey50") # lower CI
```

Hide

```
lines(future, Xs.CI[2,], col = "grey50") # upper CI
lines(future, VaR, col = "maroon3") # VaR_alpha
```

Hide

```
legend("bottomright", bty = "n", lty = rep(1, 4),
       col = c("black", "royalblue3", "grey50", "maroon3"),
       legend = c("(Aggregated) loss", "(Simulated) predicted loss",
                  "95% CIs", as.expression(substitute("Simulated"~VaR[a], list(a = alpha)))))
```



References

1. Coles SG (2001). An Introduction to Statistical Modelling of Extreme Values. Springer.
2. Hofert M, Kojadinovic I, Mächler M, & Yan J (2018). Elements of Copula Modeling with R. Springer. https://doi.org/10.1007/978-3-319-89635-9 (https://doi.org/10.1007/978-3-319-89635-9).
3. McNeil AJ, Frey R, & Embrechts P (2015). Quantitative Risk Management, Concepts, Techniques and Tools, Revised Edition. Princeton University Press.