Technische Universität München

Fakultät für Mathematik
Lehrstuhl für Mathematische Statistik

Visualization of Trivariate Vine Copulae

Bachelor's Thesis by Stefan Glogger

Supervisor: Prof. Claudia Czado, Ph.D.

Advisors: Prof. Claudia Czado, Ph.D. and Matthias Killiches

Submission Date: September 29, 2015

I hereby declare that this thesis is my own work and that no other sources have been used except those clearly indicated and referenced.

Garching, September 29, 2015

Stefan Glogger

Zusammenfassung

In dieser Arbeit visualisieren wir trivariate Copulae. Copulae stellen einen Ansatz der Statistik dar, um multivariate Datensätze zu beschreiben. Basierend auf der theoretischen Grundlage der Paar-Copula Konstruktion (PCC) bauen wir trivariate Copulae aus drei bivariaten Copulae auf. Diesen Ansatz nutzen wir zur Visualisierung von trivariaten Copulae anhand von dreidimensionalen Konturlinien, was wir mittels des statistischen Programmes *R* umsetzen.

Nach der Einführung von Notation und einem Kapitel über die ein-, zwei- und dreidimensionale Normalverteilung wird im zweiten Kapitel grundlegende Copula-Theorie vorgestellt. Das folgende Kapitel führt über die Paar-Copula Konstruktion hin zur Visualisierung trivariater Copulae. Die Umsetzung mittels *R* wird im vierten Kapitel beschrieben und es werden zahlreiche Szenarien basierend auf verschiedensten bivariaten Copulae in geordneter Art und Weise vorgestellt. Die nun zur Verfügung stehende Art der Darstellung wird abschließend praktisch eingesetzt, um die Abhängigkeiten von drei Variablen des Uranium Datensatzes zu zeigen.

Abstract

In this thesis we visualize trivariate copulae. Copulae provide one statistical way of modelling multivariate data. We construct trivariate copulae out of three bivariate copulae using the theory of pair copula construction (PCC). Based on this we visualize the three dimensional contour lines of trivariate copulae and implement this visualization with the statistical software *R*.

After introducing notation and a chapter on one, two and three dimensional normal distributions, the second chapter mentions basic theory of copulae. The following chapter on pair copula construction leads to the visualization of trivariate copulae. The implementation in *R* is covered in chapter four. Also lots of scenarios based on various bivariate copulae are provided in this chapter in a systematic manner. Finally we use this way of visualization to examine dependence among three variables out of the uranium data set.

Acknowledgement

First of all, I want to thank my supervisor Prof. Claudia Czado for giving me the opportunity for this thesis, the complaisant general conditions for it and her helpful advice during the working process. Prof. Czado has given me the possibility to enlarge my knowledge on the field of statistics and explore the theory of copulae.

Moreover I want to thank my advisor Matthias Killiches for supporting me throughout the process of this thesis. Especially his tips on writing in academic manner were very beneficial. I also would like to thank his colleague Thomas Nagler for his hints on the implementation in *R*.

Lastly, and most importantly, I am grateful for my family for their love, support, time and steady encouragement.

Contents

Introduction

Nowadays “Big Data” is a huge topic. Most people mention this to be the next big thing and its possibilities seem to be unlimited. It could be used for analysing financial transactions and discovering irregularities, for predicting epidemics or for faster and more precise market research. Whenever one wants to use data, he will first define which parameters are of interest to him and then he will collect a lot of information about them. But probably the most difficult part is how to infer facts out of this bunch of informations. Researchers are especially curious about understanding which parameters depend on each other and to what extend.

An interesting way to take care of this flood of information is to visualize data sets. One may use statistical software like *R* for this and produce barplots, pies or boxplots. And if he likes to examine dependence among multiple variables, he might prefer scatterplots, QQ plots or contour plots. These are well known concepts and can be produced using *R* and already implemented functions. By doing so, one will discover that these basic functions allow to visualize dependence between two variables. If he now wants to see also dependence among three variables, he might get stuck first and has to take some effort to handle this situation.

When examining high dimensional data sets, one might use multivariate distribution functions (like a normal distribution) to describe dependencies among multiple variables. A main limitation of this is that marginal distributions and dependence have to be modelled together. Another approach to model dependence between variables, which allows to take care of marginal distribution and dependence separately, are copulae. Its basic theorem is the Theorem of Sklar (1959), which states that for absolutely continuous random variables $X_1, \dots X_n$ with marginal distributions $F_1, \dots F_n$ and joint distribution F, there exists a unique copula C describing the joint distribution function of $X_1, \dots X_n$.

Sklar’s theorem can be used to model multidimensional dependence structures using just bivariate copulae as building blocks. This process is called “pair copula construction” (PCC). The underlying bivariate copulae can be divided in various classes (like elliptical, Archimedean or Tawn copulae) and are well known.

Basically we will use the theory of copulae and the pair copula construction to visualize dependence of three random variables, implement the whole process in *R* and have a look at lots of example scenarios. The general outline of each chapter is as follows.

First of all **Chapter 0** contains some notation which will be used throughout the thesis. A jutting position at the real beginning is reserved for this.

Chapter 1 resumes basic facts of the normal distribution function. It starts with the univariate case and makes its way over the bivariate to a trivariate normal distribution. In each dimension the probability density function and corresponding contour levels are plotted. These are used to finally take a look at the goal of this thesis in a graphical way.

Chapter 2 introduces basic theory about copulae. It starts with the definition of a copula and states Sklar's Theorem. This is applied to construct a bivariate Gaussian copula. The next section takes a look at the dependence measures Kendall's τ and Spearman's ρ. Afterwards different bivariate copula classes are shown, arranged in proper order. Elliptical copulae first, followed by Archimedean copulae. At the end of the chapter the relationship between copula parameters and Kendall's τ is stated.

The trivariate Gaussian copula is introduced in **Chapter 3**. This chapter also contains a study of pair copula constructions in three dimensions and takes the simplifying assumption into account. The chapter is concluded by defining the pair copula construction of a joint density in three dimensions.

Chapter 4 is the main part of the thesis as it contains the visualization of trivariate copulae using *R*. The chapter starts with the workflow of the implementation. Here the concepts and ideas used during programming are shown, whereas code and further explanations are given in the appendix. It continues with the description of the setting of our visualizations. The next two sections take a look at numerous visualizations of trivariate copulae constructed out of elliptical, Archimedean or Tawn copulae as building blocks. Finally the visualization tools are applied to a practical example, namely the visualization of the Uranium data set. In total we visualize 50 scenarios.

Chapter 5 gives a brief summary of the thesis. It first resumes the purpose of the thesis, followed by a recapitulation of each chapter. Finally a general conclusion is given.

The **Appendices** provide additional material for the previous chapters. Appendix A shows how the plots of probability density functions and contour levels, showed in Chapter 1, are done in *R*. The code for Chapter 2 on bivariate copulae is given in Appendix B, whereas Appendix C contains all the code for visualizing trivariate copulae as covered by Chapter 4. This code is also provided in the self written *R* package "copulaSG". Ultimately Appendix D presents an overview on bivariate copula families and their implementation in the *R* package "VineCopula". It finishes with a table listing all scenarios we have visualized.

Chapter 0

Notation

During the whole thesis we will use some special notation and introduce this right at the beginning.

Notation 0.1 We will use three different scales for random variables:

- $X \in \mathbb{R}$ The "real world" random variable distributed according to the distribution function F.
- $U \in [0,1]$ The random variable derived by $F(X)$.
- $Z \in \mathbb{R}$ The inverse given by $\Phi^{-1}(U)$.

The $\Phi(\cdot)$ mentioned above represents the standard normal cumulative distribution function (cdf) with zero mean and standard deviation of 1 and is further explained in Example 1.4.

Remark 0.2 U is a uniform $[0,1]$ distributed random variable.

Proof. $F(u) = P(U \leq u) = P(F(X) \leq u) = P(X \leq F^{-1}(u)) = F(F^{-1}(u)) = u$,
where solely the first F corresponds to the distribution of the random variable U and the latter F's correspond to the distribution function of the random variable X. □

Remark 0.3 Z is a standard normal distributed random variable.

Proof. $F(z) = P(Z \leq z) = P(\Phi^{-1}(U) \leq z) = P(U \leq \Phi(z)) = \Phi(z)$,
where F denotes the distribution of the random variable Z, and Φ stands for the standard normal distribution function, which is further described in Example 1.4. □

Notation 0.4 We will mark multidimensional objects in boldface, e.g. $\mathbf{x}$ is a multidimensional vector.

Notation 0.5 When talking about *R* code, we will use the following notation:

- **Packages** are marked in boldface,
- *Functions* are denoted in italic and
- `Parameters` are highlighted in typewriter style.

Notation 0.6 Whenever we speak of a **quantile** of a multivariate distribution or probability density function, the following is meant: We evaluate this function on a discrete number of points of an restricted domain (this is in most cases quadratic around the point of origin) and then take the corresponding quantile of these evaluations.

Chapter 1

The Normal Distribution and Other Basic Concepts of Statistics

First we want to resume some basics about normal distributions. We start with a short review of the one dimensional case and visualize its probability density function together with some contour levels. Then we dive in the two dimensional case, give some further definitions and also visualize both properties. Here we also get to know a first measure of dependence and what margins are. Afterwards we have a look at the three dimensional distribution and finish this chapter with a graphical view on what to do in this thesis.

1.1 The Univariate Normal Distribution

We start with the notation of a univariate normal distribution and define its probability density function as well as its cumulative distribution function.

Notation 1.1 We denote a one dimensional random variable X that is normal distributed with mean $\mu \in \mathbb{R}$ and variance $\sigma^2 \in \mathbb{R}_{>0}$ by $X \sim N_{\mu,\sigma^2}$.

Definition 1.2 Let $X \in \mathbb{R}$ be a univariate normal distributed random variable ($X \sim N_{\mu,\sigma^2}$). Its **probability density function** (pdf) is given by

$$f_{\mu,\sigma^2}(x) := \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right), \quad x \in \mathbb{R} . \tag{1.1}$$

Definition 1.3 Let $X \sim N_{\mu,\sigma^2}$. Its **cumulative distribution function** (cdf) is given by

$$F_{\mu,\sigma^2}(x) := P(X \leq x) = \int_{-\infty}^{x} f_{\mu,\sigma^2}(t)\,\mathrm{d}t = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{x} \exp\left(-\frac{1}{2\sigma^2}(t-\mu)^2\right) \mathrm{d}t\,, \quad x \in \mathbb{R} .$$

Even if the normal distribution is well known, we present the standard normal distribution as an example. It will be our accompanying example throughout the thesis.

Example 1.4 *Univariate standard normal distribution.*
A one dimensional normal distributed random variable X with mean $\mu = 0$ and variance $\sigma^2 = 1$ is called **standard normal distributed**. We denote this by $X \sim N_{0,1}$. We abbreviate its **probability density function** by $\phi(\cdot)$ and its **cumulative distribution function** by $\Phi(\cdot)$, i.e.

$$\Phi(x) := P(X \leq x) = \int_{-\infty}^{x} \phi(t)\,\mathrm{d}t = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}}\,\mathrm{d}t\,, \quad x \in \mathbb{R}\,.$$

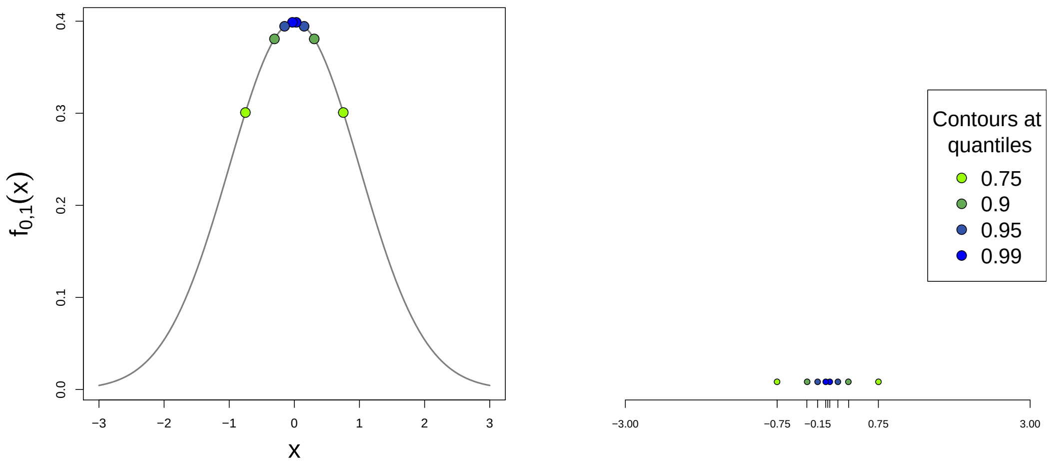

Figure 1.1: Left: Probability density function of a one dimensional standard normal distribution with contour lines at 0.75, 0.90, 0.95 and 0.99 quantiles. Right: Contour lines on their own.

The left picture of Figure 1.1 visualizes the probability density function of a one dimensional standard normal distributed random variable with added contour levels at the 0.75, 0.90, 0.95 and 0.99 quantiles. Note that the probability density function of a one dimensional distribution is two dimensional and the corresponding contour levels on their own, shown in the right picture, are one dimensional.

In the future we will need to standardize random variables. For this we use the following

Definition 1.5 (Probability integral transform). The transformation $u := F(x)$ is called **probability integral transform** (PIT).

1.2 The Bivariate Normal Distribution

For two dimensions we start as we did in the one dimensional case. We first define the probability density function of a bivariate normal distrubed random variable and its cumulative distribution function.

Notation 1.6 Let X_1, X_2 be two random variables. Let $X_1 \sim N_{\mu_1,\sigma_1^2}$ and $X_2 \sim N_{\mu_2,\sigma_2^2}$. Now we can bring them together in a vector $\mathbf{X} = (X_1, X_2)^T$, which is then **bivariate normal distributed**. We denote this by

$$\mathbf{X} \sim N_{\boldsymbol{\mu},\Sigma}$$

with mean vector $\boldsymbol{\mu} \in \mathbb{R}^2$ and covariance matrix $\Sigma \in \mathbb{R}^{2\times 2}$

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} := \begin{pmatrix} \mathrm{E}[X_1] \\ \mathrm{E}[X_2] \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{pmatrix} := \begin{pmatrix} \mathrm{cov}(X_1, X_1) & \mathrm{cov}(X_1, X_2) \\ \mathrm{cov}(X_2, X_1) & \mathrm{cov}(X_2, X_2) \end{pmatrix} .$$

Definition 1.7 Let $\mathbf{X} \in \mathbb{R}^2$ be a bivariate normal distributed random variable ($\mathbf{X} \sim N_{\boldsymbol{\mu},\Sigma}$) and Σ be not degenerated (i.e. $\det(\Sigma) \neq 0$). Its **probability density function** is given by

$$\begin{aligned} f_{\boldsymbol{\mu},\Sigma}(\mathbf{x}) &:= \frac{1}{2\pi}|\Sigma|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right) \qquad (1.2) \\ &= \frac{\exp\left(-\frac{\sigma_{22}(x_1-\mu_1)^2-\sigma_{21}(x_2-\mu_2)(x_1-\mu_1)-\sigma_{12}(x_1-\mu_1)(x_2-\mu_2)+\sigma_{11}(x_2-\mu_2)^2}{2(\sigma_{11}\sigma_{22}-\sigma_{12}\sigma_{21})}\right)}{2\pi\sqrt{\sigma_{11}\sigma_{22}-\sigma_{12}\sigma_{21}}}, \quad \mathbf{x} \in \mathbb{R}^2 . \end{aligned}$$

Definition 1.8 Let $\mathbf{X} = (X_1, X_2)^T \sim N_{\boldsymbol{\mu},\Sigma}$. Its **two dimensional normal distribution function** is given by

$$F_{\boldsymbol{\mu},\Sigma}(\mathbf{x}) := P(X \leq \mathbf{x}) = \int_{-\infty}^{x_1}\int_{-\infty}^{x_2} f_{\boldsymbol{\mu},\Sigma}(x_1, x_2)\,\mathrm{d}x_1\,\mathrm{d}x_2 .$$

As we are able to take care of two random variables now, we may also be interested in the density or distribution of just one of these variables and forget about the other. This can be done by looking on the marginals.

Definition 1.9 Let $\mathbf{X} = (X_1, X_2)^T \sim N_{\boldsymbol{\mu},\Sigma}$. The **marginal density function** of the first coordinate is given by

$$f_1(x_1) := \int_{-\infty}^{\infty} f_{\boldsymbol{\mu},\Sigma}(x_1, x_2)\,\mathrm{d}x_2 = f_{\mu_1,\sigma_{11}}(x_1)$$

and thus a one dimensional probability density function. The second coordinate works the same way.

Definition 1.10 Consequentially we get the **marginal distribution function**

$$F_1(x_1) \coloneqq F_{\boldsymbol{\mu},\Sigma}((x_1,\infty)) = F_{\mu_1,\sigma_{11}}(x_1)$$

Because of having two random variables now, we also would like to measure dependence between them. A first measure for this is the (Pearson) correlation coefficient, which is defined next. Furthermore some of its properties are mentioned. Building up on this measure of dependence we calculate the probability density function of a bivariate normal distribution.

Definition 1.11 Let $X_1,\ X_2$ be two random variables and let $\sigma_{11} \coloneqq \operatorname{Var}(X_1) > 0$ and $\sigma_{22} \coloneqq \operatorname{Var}(X_2) > 0$. The (Pearson) **correlation coefficient** between them is given by

$$\rho = \rho_{12} \coloneqq \frac{\operatorname{cov}(X_1, X_2)}{\sqrt{\operatorname{Var}(X_1)}\,\sqrt{\operatorname{Var}(X_2)}} = \frac{\sigma_{12}}{\sigma_{11}\sigma_{22}}\ ,$$

and the correlation matrix $R \in [-1,1]^{2\times 2}$ is given by

$$R \coloneqq \begin{pmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{pmatrix} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} .$$

Remark 1.12 Some properties of the (Pearson) correlation coefficient are given in Section 2.1 of Czado (2013) on page 17. These are partly mentioned here and extended at some points. So the (Pearson) correlation coefficient

- measures linear dependence with a range of $[-1,1]$,
- is invariant under strictly increasing linear transformations, but
- may change by other monotone increasing transformations of the margins and
- might depend on the marginal distributions of the underlying random variables, see Example 3.1 in Kurowicka and Cooke (2006).

Example 1.13 *Bivariate standard normal distribution with correlation.*
Let X_1, X_2 be univariate standard normal distributed random variables and let their correlation coefficient be given by ρ. Due to Definition 1.11 it holds

$$\Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{pmatrix} = \begin{pmatrix} \sigma_{11} & \rho\sigma_{11}\sigma_{22} \\ \rho\sigma_{11}\sigma_{22} & \sigma_{22} \end{pmatrix} .$$

As X_1, X_2 are standard normal distributed, we have $\mu_1 = \mu_2 = 0$ and $\sigma_{11} = \sigma_{22} = 1$. So we get for the probability density function of $\mathbf{X} = (X_1, X_2)^T$

$$\begin{aligned} f_{\boldsymbol{\mu},\Sigma}(\mathbf{x}) &\overset{1.2}{=} \frac{1}{2\pi}|\Sigma|^{-1/2}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right) \\ &= \frac{1}{2\pi}\begin{vmatrix}1 & \rho \\ \rho & 1\end{vmatrix}^{-1/2}\exp\left(-\frac{1}{2}\mathbf{x}^T\begin{pmatrix}1 & \rho \\ \rho & 1\end{pmatrix}^{-1}\mathbf{x}\right) \\ &= \frac{1}{2\pi\sqrt{1-\rho^2}}\exp\left(-\frac{x_1^2+x_2^2-2\rho x_1 x_2}{2(1-\rho^2)}\right). \end{aligned} \tag{1.3}$$

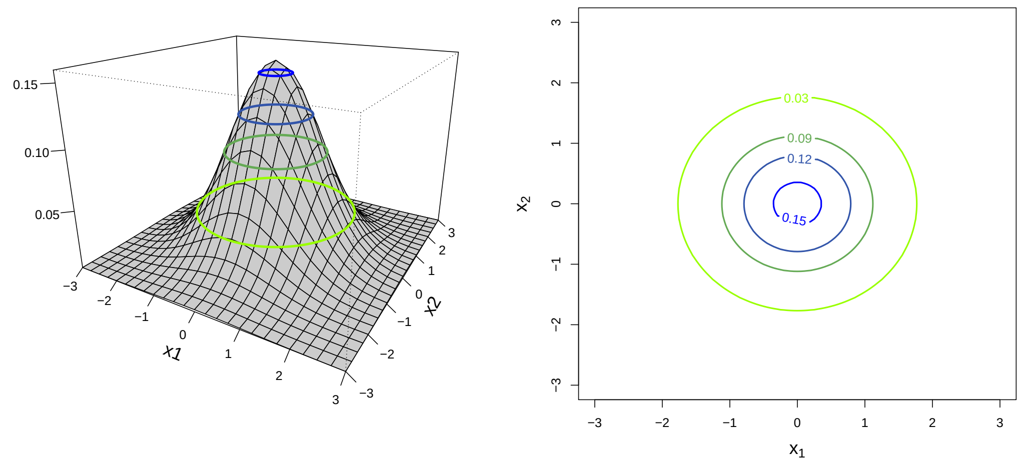

Figure 1.2: Left: Probability density function of a two dimensional standard normal distribution with zero correlation. Contour lines are added at 0.75, 0.90, 0.95 and 0.99 quantiles.
Right: Contour lines (two dimensional) on their own.

Now we visualize the probability density function of a two dimensional standard normal distribution with zero correlation in the left picture of Figure 1.2 . Again we add contour lines at the 0.75, 0.90, 0.95 and 0.99 quantiles, which are shown on their own in the right picture. Note that the probability density function is drawn in a three dimensional plot and the corresponding contour lines in a two dimensional one.

1.3 The Trivariate Normal Distribution

For the three dimensional case we start as in both sections before with notation. We continue with defining the probability density function as well as the corresponding cumulative distribution function of a trivariate normal distributed random variable.

Notation 1.14 Now we bring three normal distributed random variables X_1, X_2 and X_3 together to a trivariate random variable $\mathbf{X} = (X_1, X_2, X_3)^T \in \mathbb{R}^3$. We denote this trivariate normal distributed variable by $\mathbf{X} \sim N_{\boldsymbol{\mu},\Sigma}$ with mean vector

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix} := \begin{pmatrix} \mathrm{E}[X_1] \\ \mathrm{E}[X_2] \\ \mathrm{E}[X_3] \end{pmatrix}$$

and covariance matrix

$$\Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix} := \begin{pmatrix} \mathrm{cov}(X_1, X_1) & \mathrm{cov}(X_1, X_2) & \mathrm{cov}(X_1, X_2) \\ \mathrm{cov}(X_2, X_1) & \mathrm{cov}(X_2, X_2) & \mathrm{cov}(X_2, X_3) \\ \mathrm{cov}(X_3, X_1) & \mathrm{cov}(X_3, X_2) & \mathrm{cov}(X_3, X_3) \end{pmatrix} .$$

Definition 1.15 Let $\mathbf{X} \in \mathbb{R}^3$ be a trivariate normal distributed random variable ($\mathbf{X} \sim N_{\boldsymbol{\mu},\Sigma}$) and Σ be not degenerated (i.e. $\det(\Sigma) \neq 0$). The **probability density function** of $\mathbf{X}$ is then given by

$$f_{\boldsymbol{\mu},\Sigma}(\mathbf{x}) := \frac{1}{(2\pi)^{\frac{3}{2}}} |\Sigma|^{-\frac{1}{2}} \exp\Big(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \Big) , \mathbf{x} \in \mathbb{R}^3 .$$

Definition 1.16 Let $\mathbf{X} \sim N_{\boldsymbol{\mu},\Sigma}$. Its **three dimensional normal distribution function** is given by

$$F_{\boldsymbol{\mu},\Sigma}(\mathbf{x}) := P(X \leq \mathbf{x}) = \int\limits_{-\infty}^{x_1} \int\limits_{-\infty}^{x_2} \int\limits_{-\infty}^{x_3} f_{\boldsymbol{\mu},\Sigma}(x_1, x_2, x_3) \, \mathrm{d}x_1 \, \mathrm{d}x_2 \, \mathrm{d}x_3 .$$

As we have three variables, we may want to take a look on how one single variable behaves while disregarding the other variables. Thus we study the margins once again.

Definition 1.17 Let $\mathbf{X} \sim N_{\boldsymbol{\mu},\Sigma}$. The **marginal density function** of the first coordinate is given by

$$f_1(x_1) := \int\limits_{-\infty}^{\infty} \int\limits_{-\infty}^{\infty} f_{\boldsymbol{\mu},\Sigma}(x_1, x_2, x_3) \, \mathrm{d}x_2 \, \mathrm{d}x_3 = f_{\mu_1,\sigma_{11}}(x_1)$$

and thus a one dimensional probability density function. The other coordinates work the same way.

Definition 1.18 Consequentially we get the **marginal distribution function** by

$$F_1(x_1) := F_{\boldsymbol{\mu},\Sigma}((x_1, \infty, \infty)) = F_{\mu_1,\sigma_{11}}(x_1) .$$

We close this section with one remark on multidimensional distributions.

Remark 1.19 A random variable $\mathbf{X} = (X_1, \ldots, X_d)^T \in \mathbb{R}^d$ which is **multivariate normal distributed** with mean vector $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance matrix $\Sigma \in \mathbb{R}^{d\times d}$ is denoted by $\mathbf{X} \sim N_{\boldsymbol{\mu},\Sigma}$. Its probability density function is given by

$$f_{\boldsymbol{\mu},\Sigma}(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{d}{2}}} |\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right) .$$

1.4 Graphical Exploration of Thesis Goal

Let us take a look on what we have done so far. We started with a univariate normal distribution and visualized the probability density function of a univariate standard normal distribution in Figure 1.1 . We also calculated some quantiles of the plotted region and added the corresponding contour lines to the plot. Next to this plot we positioned another plot just displaying the contour lines. To summarize the univariate case: The plot about the probability density function was *two* dimensional and the one of the contour lines was *one* dimensional.

Then we took over to a bivariate normal distribution and visualized the probability density function of a bivariate standard normal distribution with zero correlation in Figure 1.2 . Again we added contour lines of some quantiles and also displayed just the contour lines in another plot. To summarize the bivariate case: The plot about the probability density function was *three* dimensional and the one of the contour lines was *two* dimensional.

If we had continued with this procedure for the trivariate case, we would have drawn the probability density function in a *four* dimensional plot and the one of the contour lines in a *three* dimensional one.

The rest of this thesis is about a way to visualize three dimensional copulae. The definition of a copula is part of the next chapter, but one can look at it as a three dimensional distribution function. As we want to visualize them, it should be clear at this point why we will image them quite indirectly via contour lines. The direct way of showing the probability density function would result in a four dimensional plot.

Chapter 2

Copula Theory

In this chapter we want to introduce some copula theory. We first define a copula and state the fundamental theorem about copulae, named “Sklar’s Theorem”. We show directly how to use this new theory by studying a bivariate Gaussian copula. This is followed by the definition of two dependence measures as different approaches to the (Pearson’s) correlation coefficient. Afterwards we show a variety of bivariate copulae classified in elliptical copulae, Archimedean copulae with one or two parameters and extreme value copulae. At the end we mention the relationship between copula parameters and one dependence measure.

The whole chapter is based on Czado (2013) with changes in wording and presentation of the ideas.

2.1 Definition of Copula and Sklar’s Theorem

We start this basic chapter on copula theory with the general definition of a copula and Sklar’s Theorem, which we then apply to get a bivariate Gaussian copula.

Definition 2.1 A d dimensional **copula** C is a multivariate distribution function on the d dimensional hyper cube $[0,1]^d$ with uniformly distributed marginals. We denote the corresponding **copula density** by c and it can be obtained by partial differentiation (if this is possible), i.e.

$$c(u_1, \dots, u_d) := \frac{\partial^d}{\partial u_1 \dots \partial u_d} C(u_1, \dots, u_d) \ .$$

Theorem 2.2 (Sklar’s Theorem). Let $\mathbf{X}$ be a d dimensional random variable with joint distribution function F and marginal distribution functions F_i, $i = 1, \dots, d$. Then there is a d dimensional copula C with uniform marginals such that

$$F(x_1, \dots, x_d) = C(F_1(x_1), \dots, F_d(x_d)) \ .$$

In this thesis we will only be confronted with absolutely continuous distributions. For these the copula C is unique and its density c exists and one derives

$$f(x_1, \dots, x_d) = c(F_1(x_1), \dots, F_d(x_d)) f_1(x_1) \cdot \ldots \cdot f_d(x_d) \ . \qquad \text{(Sklar)}$$

Going the other way round is also possible. The copula corresponding to a multivariate distribution is given by

$$C(u_1, \dots, u_d) = F(F_1^{-1}(u_1), \dots, F_d^{-1}(u_d)) \ , \tag{2.1}$$

$$c(u_1, \dots, u_d) = \frac{f(F_1^{-1}(u_1), \dots, F_d^{-1}(u_d))}{f_1(F_1^{-1}(u_1)) \cdot \ldots \cdot f_d(F_d^{-1}(u_d))} \ . \tag{2.2}$$

Example 2.3 *Bivariate Gaussian copula.*
The bivariate Gaussian copula can be constructed using a bivariate normal distribution with zero mean and correlation ρ and applying Sklar's Theorem 2.1 to give

$$C(u_1, u_2; \rho) = \Phi_2\Big(\Phi^{-1}(u_1), \Phi^{-1}(u_2); \rho\Big) \ ,$$

where $\Phi(\cdot)$ is the distribution function of a standard normal $N_{0,1}$ distribution and $\Phi_2(\cdot, \cdot; \rho)$ denotes the bivariate normal distribution function with zero mean and correlation ρ. The corresponding copula density is given by

$$\begin{aligned}
c(u_1, u_2) &\overset{2.2}{\underset{1.9}{=}} \frac{\phi_2(\Phi^{-1}(u_1), \Phi^{-1}(u_2); \rho)}{\phi(\Phi^{-1}(u_1))\phi(\Phi^{-1}(u_2))} \\
&= \frac{\phi_2(z_1, z_2; \rho)}{\phi(z_1)\phi(z_2)} \\
&\overset{1.3}{\underset{1.1}{=}} \frac{\frac{1}{2\pi\sqrt{1-\rho^2}} \exp\Big(-\frac{z_1^2+z_2^2-2\rho z_1 z_2}{2(1-\rho^2)}\Big)}{\frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{1}{2} z_1^2\Big) \frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{1}{2} z_2^2\Big)} \\
&= \frac{1}{\sqrt{1-\rho^2}} \exp\left(-\frac{z_1^2 + z_2^2 - 2\rho z_1 z_2}{2(1-\rho^2)} + \frac{z_1^2 + z_2^2}{2}\right) \\
&= \frac{1}{\sqrt{1-\rho^2}} \exp\left(\frac{2\rho z_1 z_2 - \rho^2(z_1^2 + z_2^2)}{2(1-\rho^2)}\right) \ ,
\end{aligned}$$

where $\phi_2(\cdot, \cdot; \rho)$ is the probability density function of a bivariate standard normal distribution with correlation ρ and $\phi(\cdot)$ denotes the probability density function of a univariate standard normal distributions. In the second line we did the substitution $z_i \coloneqq \Phi^{-1}(u_i)$, $i \in \{1, 2\}$ (remember the Z scale).

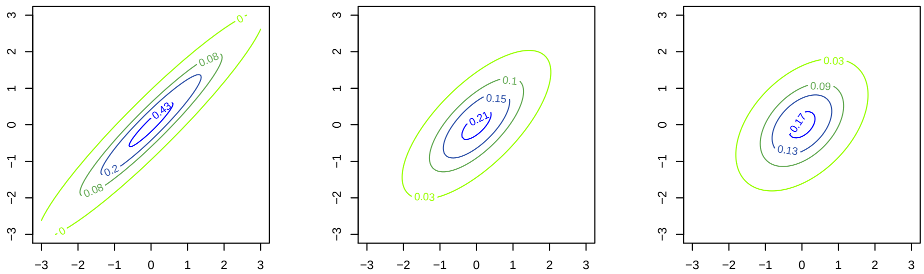

Figure 2.1: Gaussian copula

	left	mid	right
τ	0.80	0.50	0.30
θ_1	0.95	0.71	0.45

Figure 2.1 shows the contour plots of a bivariate Gaussian copula with descending τ values (from 0.8 in the left picture over 0.5 in the middle one to 0.3 in the right picture). The dependence measure Kendall's τ is explained in the following section. The levels are based on the 0.75, 0.90, 0.95 and 0.99 quantiles of the plotted area. We see the classical elliptic shape, which is the more pulled apart the higher the τ value is. The z_1 axis is drawn on the right axis and z_2 on the upper one.

The shape from left bottom to right top in each picture shows **positive dependence** (if the value of the first variable gets high, it is likely that the value of the other variable also gets high). All of the following copulae will show positive dependence. One can reach negative dependence by rotating copulae. Details on that are given in Section 3.5 in Czado (2013) on page 30.

2.2 Dependence Measures

In Definition 1.11 we already defined (Pearson's) correlation coefficient as a first measure of dependence. But we faced some kind of "bad" properties (see Remark 1.12). Now we want to introduce two other dependence measures, connect them to copulae and look at some of their nice properties, especially for usage with copulae.

Definition 2.4 Let X_1 and X_2 be two random variables. Then **Kendall's τ** is defined as the difference between the probability of concordance and the probability of discordance, i.e.

$$\tau(X_1, X_2) := P((X_{11} - X_{21})(X_{12} - X_{22}) > 0) - P((X_{11} - X_{21})(X_{12} - X_{22}) < 0) ,$$

where (X_{11}, X_{12}) and (X_{21}, X_{22}) are independent and identically distributed copies of (X_1, X_2).

Definition 2.5 Let X_1 and X_2 be two random variables with marginal distributions F_1 and F_2. Then **Spearman's $\boldsymbol{\rho_s}$** or the **rank correlation** is defined as the Pearson correlation of the random variables $F_1(X_1)$ and $F_2(X_2)$ (remember the U scale), i.e.

$$\rho_s := \rho_s(X_1, X_2) = \operatorname{corr}(F_1(X_1), F_2(X_2)) \ .$$

Theorem 2.6 Kendall's τ and Spearman's ρ_s only depend on the copula and can be expressed as follows

$$\begin{aligned} \tau &= 4 \int_{[0,1]^2} C(u_1, u_2) dC(u_1, u_2) - 1 \ , \\ \rho_s &= 12 \int_{[0,1]^2} C(u_1, u_2) du_1 du_2 - 3 \ . \end{aligned}$$

Now we get to the properties of both dependence measures which are stated in Section 2.2 in Czado (2013) on page 18. The corresponding properties of the (Pearson) correlation coefficient are given in Remark 1.12.

Remark 2.7 Kendall's τ and Spearman's ρ

- measure general dependence with a range of $[-1, 1]$,
- are invariant with respect to monotone transformations of the margins (because they are rank-based),
- do not depend on marginal distributions (as they can be expressed solely in terms of the copulae) and
- often can be fully expressed in terms of the copula parameters (see Theorem 2.16).

Another interesting aspect when talking about dependence is the probability of joint occurrence of extreme observations, meaning if one variable takes a low value, the other will also tend to be low and the same for large values. We call this **tail dependence**.

Definition 2.8 Let $\mathbf{X}$ be a two dimensional random variable with bivariate distribution F and marginal distributions F_1, F_2. The corresponding copula shall be given by C. Then the **lower tail dependence coefficient** is defined as

$$\begin{aligned} \lambda^l &:= \lim_{t \to 0^+} P\Big(X_2 \leq F_2^{-1}(t) | X_1 \leq F_1^{-1}(t)\Big) \\ &= \lim_{t \to 0^+} \frac{P\Big(X_2 \leq F_2^{-1}(t), X_1 \leq F_1^{-1}(t)\Big)}{P\Big(X_1 \leq F_1^{-1}(t)\Big)} \\ &= \lim_{t \to 0^+} \frac{C(t,t)}{t} \ , \end{aligned}$$

whereas the **upper tail dependence coefficient** is defined as

$$
\begin{aligned}
\lambda^u &:= \lim_{t \to 1^-} P\Big(X_2 > F_2^{-1}(t) | X_1 > F_1^{-1}(t)\Big) \\
&= \lim_{t \to 1^-} \frac{1 - P\Big(X_2 \leq F_2^{-1}(t)\Big) - P\Big(X_1 \leq F_1^{-1}(t)\Big) + P\Big(X_2 > F_2^{-1}(t), X_1 > F_1^{-1}(t)\Big)}{1 - P\Big(X_1 \leq F_1^{-1}(t)\Big)} \\
&= \lim_{t \to 1^-} \frac{1 - 2t + C(t,t)}{1-t} \; .
\end{aligned}
$$

Example 2.9 *Tail dependence for bivariate Gaussian copula.*
One can show that a bivariate Gaussian copula with (Pearsson) correlation ρ has either no tail dependence at all or full one:

$$
\lambda^l = \lambda^u = \begin{cases} 0 & \text{if and only if } |\rho| < 1 \\ 1 & \text{if and only if } |\rho| = 1 \end{cases} \; .
$$

2.3 Bivariate Copula Classes

2.3.1 Construction of Bivariate Copula Classes

Copulae can be constructed using different approaches. Two major principles lead to elliptical and Archimedean copulae. If one does a probability integral transform (Definition 1.5) for each margin of known multivariate elliptical distributions (like Gaussian or Student t), he will end up with so called elliptical copulae. Using generator functions leads to Archimedean copulae.

2.3.2 Elliptical Copulae

The class of elliptical copulae contains for example the bivariate Gaussian copula, which we have already seen in Example 2.3 . Now another member of this class is given.

Example 2.10 *Bivariate t copula.*
A bivariate t copula is based on the Student t distribution. If one likes to refresh his knowledge on that distribution, he should take a look in basic literature on statistics such as Hogg et al. (2013). Details on the construction of a bivariate Student t copula can be found in Czado (2013) (Example 1.10 on page 9). For this thesis, we are just interested in its probability density function and only give a short overview of the corresponding construction.

The underlying distribution is a bivariate t distribution with ν degrees of freedom, denoted by $T_\nu(\cdot)$. Furthermore it is standard t, meaning it has zero mean and correlation ρ. The probability of a univariate standard t distribution is denoted by $f_\nu(\cdot)$.

Now the copula density of a bivariate t copula is given as

$$c(u_1, u_2; \rho, \nu) = \frac{\Gamma\left(\frac{\nu+2}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{(\pi\nu)^2(1-\rho^2)}} \left(1 + \frac{z_1^2 - 2\rho z_1 z_2 + z_2^2}{\nu(1-\rho^2)}\right)^{-\frac{\nu+2}{2}} \frac{1)}{f_\nu(z_1) f_\nu(z_2)} \ ,$$

where $z_i := T_\nu^{-1}(u_i), \ \ i \in \{1, 2\}$ and $\Gamma(\cdot)$ is the gamma function.

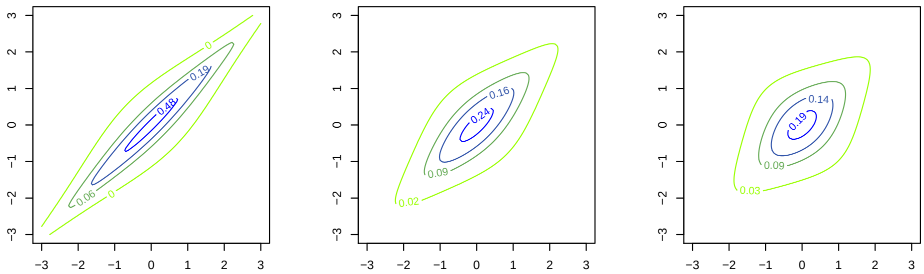

Figure 2.2: Student t copula

	left	mid	right
τ	0.80	0.50	0.30
θ_1	0.95	0.71	0.45
θ_2	3.00	3.00	3.00

The contour plots for bivariate Student t copulae, shown in Figure 2.2, have that typical rectangular shape with rounded corners. These rectangular shapes get more tightened the higher the τ value is, showing the higher dependence among the variables. Upper and lower tail dependence are equal and positive, even for negative and zero correlations, as shown in Demarta and McNeil (2005) on page 5.

2.3.3 Archimedean Copulae

We just want to highlight the most important facts about Archimedean copulae and show those members of this class that are implemented in the *R* package **VineCopula**. More details on Archimedean copulae can be found in Nelsen (2006).

Definition 2.11 (Bivariate Archimedean copulae). Let $\psi \in \Omega$ with Ω the set of all continuous, strict monotone decreasing and convex functions $g : [0, 1] \to [0, \infty]$ with $g(1) = 0$. Then a so called **Archimedean copula** with **generator** ψ is given by

$$C(u_1, u_2) = \psi^{[-1]}(\psi(u_1) + \psi(u_2)) \ .$$

The generator is called **strict** for $\psi(0) = \infty$.

Remark 2.12 $\psi^{[-1]}$ is the **pseudo-inverse** of ψ and is defined as

$$\psi^{[-1]} : [0,\infty] \to [0,1]$$
$$\psi^{[-1]}(t) := \begin{cases} \psi^{-1}(t) & , 0 \le t \le \psi(0) \\ 0 & , \psi(0) < t \le \infty \ . \end{cases}$$

Lemma 2.13 (Density of Archimedean copulae). Let C be an absolutely continuous Archimedean copula and its generator ψ be twice differentiable. Then the corresponding density is given by

$$c(u_1,u_2) = -\frac{\partial^2 C(u_1,u_2)}{\partial u_1 \partial u_2} = -\frac{\psi''(C(u_1,u_2))\psi'(u_1)\psi'(u_2)}{[\psi'(C(u_1,u_2))]^3} \ .$$

After this short theory on Archimedean copulae we will now show examples of this class that are implemented in the *R* package **VineCopula**. We divide these examples in Archimedean copulae with one parameter and those with two parameters. For which values one gets independence or full dependence and what other copula families could be obtained, is first taken from Czado (2013) and completed with Joe (1997).

Example 2.14 *Bivariate Archimedean copulae with a single parameter.*

- **Clayton copula**

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

where dependence is controlled by $\delta > 0$. Full dependence is achieved if $\delta \to \infty$ and independence if $\delta \to 0$. We define $C(0,0) := 0$.

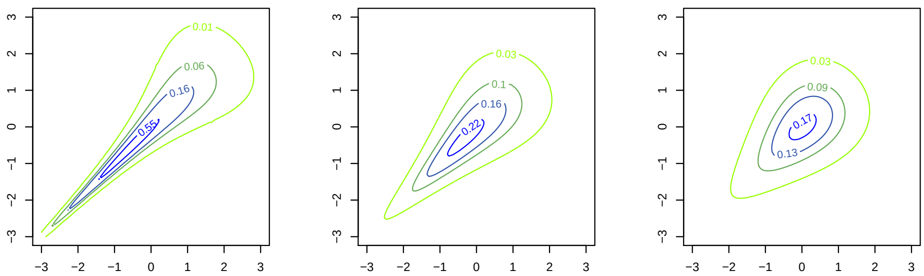

Figure 2.3: Clayton copula

	left	mid	right
τ	0.80	0.50	0.30
θ_1	8.00	2.00	0.86

In the contour plots for the Clayton copula, shown in Figure 2.3, one sees tightened contour lines on the lower left hand side (especially when focussing on the left picture with highest dependence) and fanned out ones on the upper right hand side. So the Clayton copula is best suited for modelling high lower tail dependence and weak upper tail dependence. Note that this asymmetric dependence can't be replicated by a Gaussian or t copula.

- **Gumbel copula**

$$C(u_1, u_2) = \exp\left[-\left((-\ln u_1)^\delta + (-\ln u_2)^\delta\right)^{\frac{1}{\delta}}\right] ,$$

where $\delta \geq 1$ controls dependence. One gets full dependence if $\delta \to \infty$ and independence if $\delta = 1$.

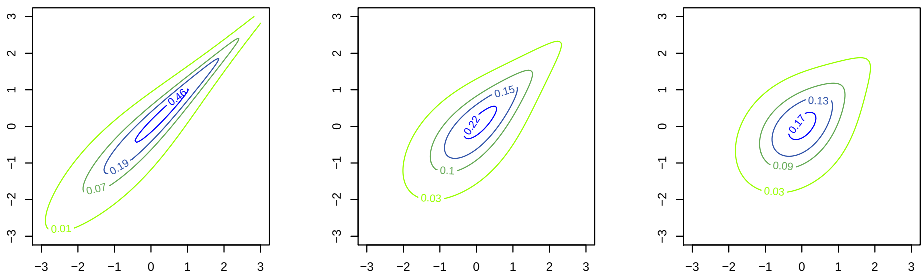

Figure 2.4: Gumbel copula

	left	mid	right
τ	0.80	0.50	0.30
θ_1	5.00	2.00	1.43

Contour plots for the Gumbel copula in Figure 2.4 are in some kind reverse to those for the Clayton copula. The Gumbel copula is suited for modelling high upper tail dependence and weaker lower tail dependence. But the difference in the values of upper and lower tail dependence is not as high as for the Clayton copula, as contour lines do not fan out that much.

- **Frank copula**

$$C(u_1, u_2) = -\frac{1}{\delta} \ln\left(1 - \frac{\left(1 - e^{-\delta u_1}\right)\left(1 - e^{-\delta u_2}\right)}{1 - e^{-\delta}}\right),$$

where $\delta \in \mathbb{R} \setminus \{0\}$. Full dependence is achieved for $\delta \to \infty$ and independence for $\delta \to 0$.

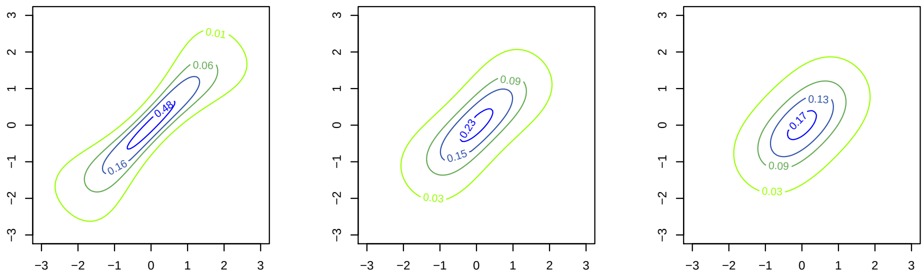

Figure 2.5: Frank copula

	left	mid	right
τ	0.80	0.50	0.30
θ_1	18.19	5.74	2.92

Figure 2.5 shows the interesting behaviour of the Frank copula. When dependence rises, contour lines are tightened together near the source of origin. This is an indicator that a Frank copula is able to model situations with strong central dependency and weak tail dependence (even better than a Gaussian copula can do). This fact is also supported when looking at the value of the contour lines. The 0.99 quantile of a Gumbel copula is with a value of 0.48 higher than the 0.43 of the corresponding Gaussian copula. But for the other quantiles the value of the Gaussian copula is higher. To sum this up, a lot of probability mass is concentrated in the centre of a Frank copula.

Another interesting fact is its radial symmetry. In fact, the family of Frank copulae is the only family among Archimedean copulae with radial symmetry, as shown in Frank (1979).

- **Joe copula**

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

where $\delta \geq 1$. One gets full dependence for $\delta \to \infty$ and the independence copula for $\delta = 1$.

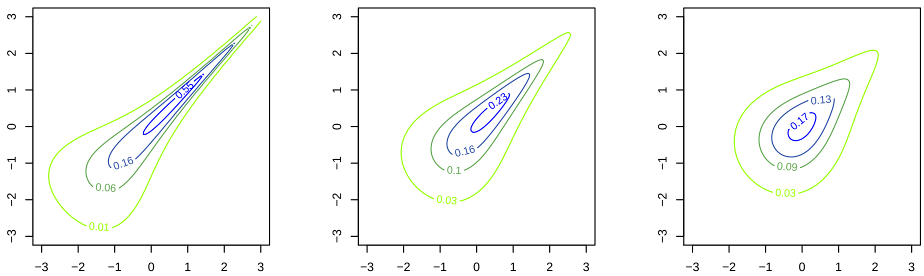

Figure 2.6: Joe copula

	left	mid	right
τ	0.80	0.50	0.30
θ_1	8.77	2.86	1.77

Finally the Joe copula, whose contour lines are shown in Figure 2.6, is similar to the Gumbel copula, but the upper tail dependence is higher. This can be seen in the tighter contour lines on the upper right hand side. So a Joe copula is closer to be the reverse of the Clayton copula than a Gumbel one.

Now we get to Archimedean copulae with two parameters. The BB notation for those was introduced in Section 5.2 in Joe (1997) on page 149 et seqq. and the following examples are taken from this book with changes in notation.

Example 2.15 *Bivariate Archimedean copulae with two parameters.*

- **BB1 copula**: Clayton-Gumbel

$$C(u_1, u_2; \theta, \delta) = \left(1 + \left[\left(u_1^{-\theta} - 1\right)^{\delta} + \left(u_2^{-\theta} - 1\right)^{\delta}\right]^{\frac{1}{\delta}}\right)^{-\frac{1}{\theta}}$$
$$= \eta\Big(\eta^{-1}(u_1) + \eta^{-1}(u_2)\Big) \;,$$

where $\theta > 0$, $\delta \geq 1$ and $\eta(s) = \eta_{\theta,\delta}(s) = \left(1 + s^{\frac{1}{\delta}}\right)^{-\frac{1}{\theta}}$. Full dependence is achieved if $\theta \to \infty$ or if $\delta \to \infty$ and independence if $\theta \to 0$ and $\delta = 1$.

A Gumbel copula is obtained for $\theta \to 0$ and a BB1 copula with $\delta = 1$ coincides with a Clayton copula.

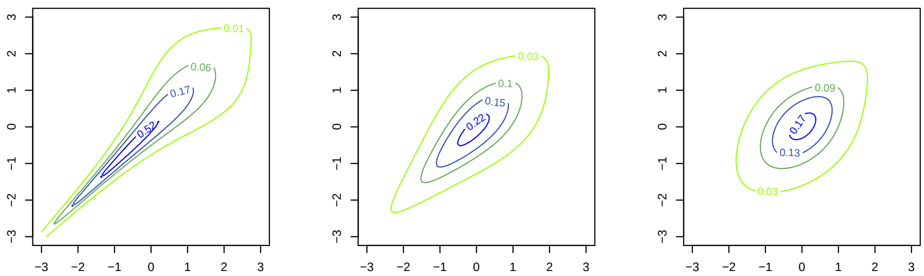

Figure 2.7: BB1 copula

	left	mid	right
τ	0.80	0.50	0.30
θ_1	5.69	1.08	0.20
θ_2	1.30	1.30	1.30

For Figure 2.7 we set the second parameter to 1.3 and the similarity to a Clayton copula can be seen (compare with Figure 2.3). But now lower tail dependence decreased a bit, whereas upper tail dependence increased and even shows up for low τ values. Furthermore upper tail dependence is invariant under changes of the first parameter. Nevertheless a BB1 copula models high lower tail dependence together with some upper tail dependence.

- **BB6 copula**: Joe-Gumbel

$$C(u_1, u_2; \theta, \delta) = 1 - \left(1 - \exp\left(-\left[(-\ln(1-(1-u_1)^\theta))^\delta + (-\ln(1-(1-u_2)^\theta))^\delta\right]^{\frac{1}{\delta}}\right)\right)^{\frac{1}{\theta}}$$
$$= \eta\left(\eta^{-1}(u_1) + \eta^{-1}(u_2)\right),$$

where $\theta \geq 1$, $\delta \geq 1$ and $\eta(s) = \eta_{\theta,\delta}(s) = 1 - \left[1 - \exp\left(-s^{\frac{1}{\delta}}\right)\right]^{\frac{1}{\theta}}$. Full dependence is achieved for $\theta \to \infty$ or $\delta \to \infty$.

One obtains a Gumbel copula for $\theta = 1$ and a Joe copula for $\delta = 1$.

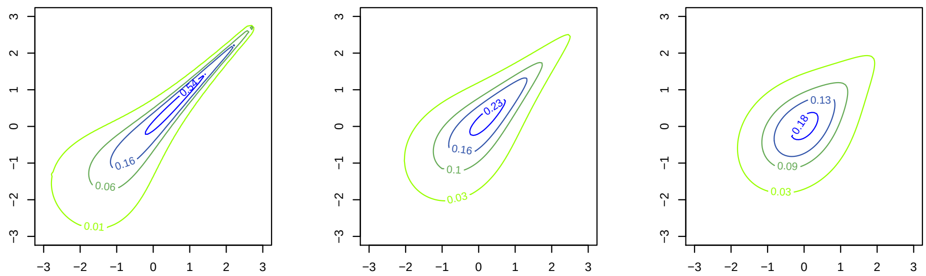

Figure 2.8: BB6 copula

	left	mid	right
τ	0.80	0.50	0.30
θ_1	6.48	1.98	1.17
θ_2	1.30	1.30	1.30

Figure 2.8 visualizes the contour lines of a BB6 copula. Lower tail dependence can not be observed (and is in fact equal to 0), whereas upper tail dependence is high. The shape is similar to that of a Joe copula.

- **BB7 copula**: Joe-Clayton

$$C(u_1, u_2; \theta, \delta) = 1 - \left(1 - \left[(1-(1-u_1)^\theta)^{-\delta} + (1-(1-u_2)^\theta)^{-\delta} - 1\right]^{-\frac{1}{\delta}}\right)^{\frac{1}{\theta}}$$
$$= \eta\Big(\eta^{-1}(u_1) + \eta^{-1}(u_2)\Big),$$

where $\theta \geq 1$, $\delta > 0$ and $\eta(s) = \eta_{\theta,\delta}(s) = 1 - \left[1 - (1+s)^{-\frac{1}{\delta}}\right]^{\frac{1}{\theta}}$.

One gets full dependence for $\theta \to \infty$ or $\delta \to \infty$. For $\delta \to 0$ one obtains the family of Joe copulae.

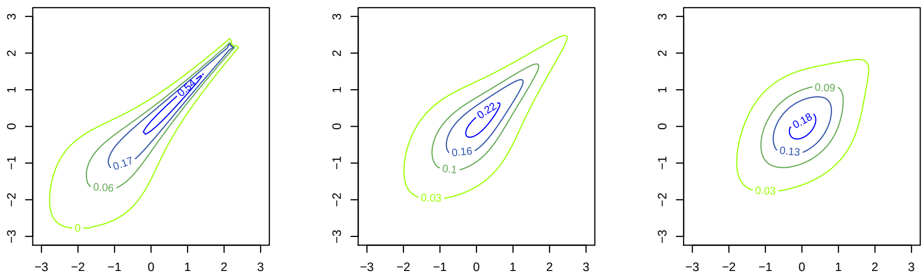

Figure 2.9: BB7 copula

	left	mid	right
τ	0.80	0.50	0.30
θ_1	8.43	2.50	1.42
θ_2	0.40	0.40	0.40

Contour lines of a BB7 copula are shown in Figure 2.9 . In our case, with a second parameter equal to 0.4, we have strong upper tail dependence. In fact upper tail dependence is independent of the second parameter and the lower one is independent of the first parameter. That makes the BB7 copula flexible in usage.

- **BB8 copula**: Joe-Frank

$$C(u_1, u_2; \theta, \delta) = \delta^{-1}\left(1 - \left[1 - \left(1-(1-\delta)^{\theta}\right)^{-1}\left(1-(1-\delta u_1)^{\theta}\right)\left(1-(1-\delta u_2)^{\theta}\right)\right]^{\frac{1}{\theta}}\right),$$

where $\theta \geq 1, 0 < \delta \leq 1$. Independence is achieved for $\theta = 1$ or $\delta = 0$.

One obtains a Frank copula for $\theta \to \infty$ and a Joe copula when $\delta = 1$.

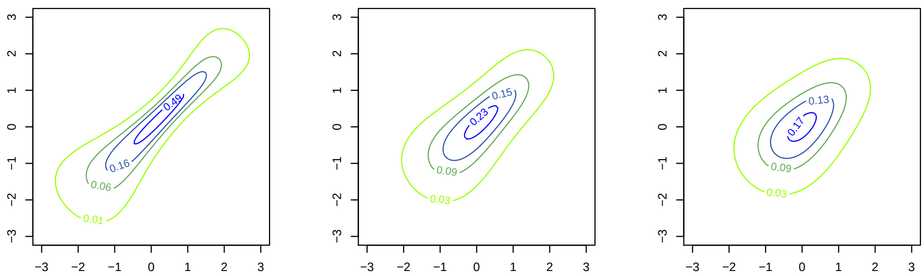

Figure 2.10: BB8 copula

	left	mid	right
τ	0.80	0.50	0.30
θ_1	16.91	5.58	3.20
θ_2	0.70	0.70	0.70

The contour lines of a BB8 copula with high value of θ, shown in the left picture of Figure 2.10, are similar to those of a Frank copula given in Figure 2.5. One also observes similarity to the Joe copula for $\delta \to 1$.

2.3.4 Extreme Value Copulae

Up to now copulae were symmetric, but we use two more bivariate copulae as building blocks, namely "Tawn type 1" and "Tawn type 2". A Tawn copula is an extreme value copula and usually has three parameters. Both copulae mentioned above are reduced to two parameters and thus special members of the family of Tawn copulae. They are implemented in the *R* package **VineCopula** and details on further characteristics can be found in its manual or in Eschenburg (2013). We just show their asymmetric contour lines here.

- **Tawn type 1 copula**

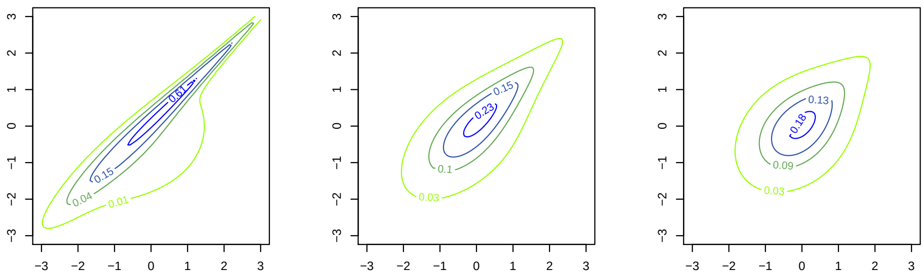

Figure 2.11: Tawn type 1 copula

	left	mid	right
τ	0.80	0.50	0.30
θ_1	8.28	2.16	1.47
θ_2	0.90	0.90	0.90

Contour lines of a Tawn type 1 copula, as given in Figure 2.11, are similar to those of a Gumbel copula (compare to Figure 2.4). But for high dependence as in the left picture a bulge at the lower side gets visible.

- **Tawn type 2 copula**

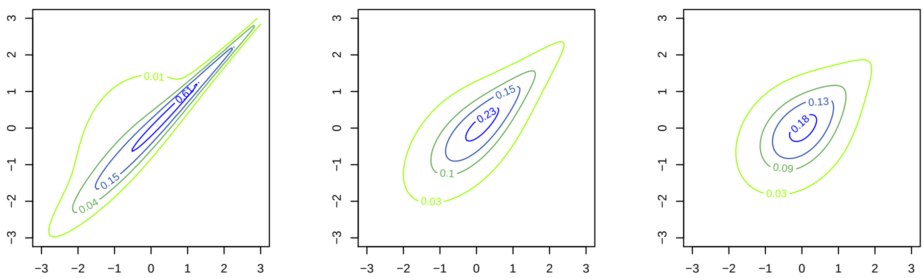

Figure 2.12: Tawn type 2 copula

	left	mid	right
τ	0.80	0.50	0.30
θ_1	8.28	2.16	1.47
θ_2	0.90	0.90	0.90

Figure 2.12 contains contour lines of a Tawn type 2 copula, which are similar to those of a Tawn type 1 copula. The only difference is that the bulge is now to the upper side.

2.4 Relationship between Kendall's τ and Copula Parameters

Now we want to express Kendall's τ solely in terms of the copula parameters as stated in Remark 2.7 .

Theorem 2.16 (Kendall's τ for bivariate elliptical and Archimedean copulae).

- For an elliptical copula the following relationship between Kendall's τ and correlation ρ holds
$$\tau = \frac{2}{\pi} \sin(\rho)$$
- For an Archimedean copula with generator ψ the corresponding Kendall's τ satisfies
$$\tau = 1 + 4 \int_0^1 \frac{\psi(t)}{\psi'(t)} \, \mathrm{d}t \; .$$

The proof for elliptical copulae can be found in Lindskog et al. (2002) and the one for Archimedian copulae is given in the proof for Corollary 5.1.4 in Nelsen (2006) on page 163.

Chapter 3

Trivariate Copulae

In this chapter we want to discuss trivariate copulae. We discuss a trivariate Gaussian copula as a first example. Then we show how to get trivariate copulae by pair copula construction, which we use later on for visualization.

3.1 Trivariate Gaussian Copula

Example 3.1 *Trivariate Gaussian copula.*
After applying the inverse of Sklar's theorem to a trivariate Gaussian distribution with zero mean and symmetric positive definite correlation matrix $R \in [-1,1]^{3\times 3}$ (Definition 1.11), we end up with the trivariate Gaussian copula

$$C(\mathbf{u}; R) = \Phi_3\Big(\Phi^{-1}(u_1), \Phi^{-1}(u_2), \Phi^{-1}(u_3); R\Big) ,$$

where $\Phi(\cdot)$ is the cumulative distribution function of a standard normal distribution and $\Phi_3(\cdot,\cdot,\cdot;R)$ denotes the trivariate standard normal distribution function with zero mean and correlation matrix R.

The corresponding copula density can be derived similar as in Example 2.3 and one gets

$$c(\mathbf{u}; R) = |R|^{-\frac{1}{2}} \exp\bigg(\frac{1}{2}\mathbf{z}^T (I_3 - R^{-1})\mathbf{z}\bigg),$$

where $\mathbf{z} = (z_1, z_2, z_3)^T \in \mathbb{R}^3$ with $z_i := \Phi^{-1}(u_i)$, $i \in \{1,2,3\}$ (remember the Z scale) and I_3 is the three dimensional identity matrix.

3.2 Pair Copula Construction

As we want to visualize three dimensional copulae, we have to construct them first. Using the inverse of Sklar's theorem, like we did in the example above, worked fine for a trivariate Gaussian copula. But there is another, more convenient way named "pair copula construction". Roughly spoken, we express the three dimensional density function using only bivariate building blocks. The main tool will be conditioning. These approaches were developed in Joe (1996), Bedford and Cooke (2001) and Bedford and Cooke (2002).

In the following we take a trivariate probability density function $f(\cdot,\cdot,\cdot) = f_{123}(\cdot,\cdot,\cdot)$ and split it up by conditioning. To visualize which parts belong together, but not attracting too much attention, we use unobtrusive colours.

$$
\begin{aligned}
f(x_1,x_2,x_3) &= \frac{f_{123}(x_1,x_2,x_3)}{f_{12}(x_1,x_2)}\frac{f_{12}(x_1,x_2)}{f_1(x_1)}f_1(x_1) = f_{3|12}(x_3|x_1,x_2)f_{2|1}(x_2|x_1)f_1(x_1)\\
f_{2|1}(x_2|x_1) &= \frac{f_{12}(x_1,x_2)}{f_1(x_1)} \overset{Sklar}{=} \frac{c_{12}(F_1(x_1),F_2(x_2))f_1(x_1)f_2(x_2)}{f_1(x_1)}\\
&= c_{12}(F_1(x_1),F_2(x_2)f_2(x_2)\\
f_{13|2}(x_1,x_3|x_2) &\overset{Sklar}{=} c_{13;2}(F_{1|2}(x_1|x_2),F_{3|2}(x_3|x_2);x_2)f_{1|2}(x_1|x_2)f_{3|2}(x_3|x_2)\\
f_{3|12}(x_3|x_1,x_2) &= \frac{f_{123}(x_1,x_2,x_3)}{f_{12}(x_1,x_2)} = \frac{f_{13|2}(x_1,x_3|x_2)f_2(x_2)}{f_{12}(x_1,x_2)} = \frac{f_{13|2}(x_1,x_3|x_2)}{f_{1|2}(x_1|x_2)}\\
&= \frac{c_{13;2}(F_{1|2}(x_1|x_2),F_{3|2}(x_3|x_2);x_2)f_{1|2}(x_1|x_2)f_{3|2}(x_3|x_2)}{f_{1|2}(x_1|x_2)}\\
&= c_{13;2}(F_{1|2}(x_1|x_2),F_{3|2}(x_3|x_2);x_2)f_{3|2}(x_3|x_2)\\
f_{3|2}(x_3|x_2) &= c_{23}(F_2(x_2),F_3(x_3))f_3(x_3)
\end{aligned}
$$

Here two ways of conditioning occur. $f_{i|j}(\cdot,x_j)$ denotes the conditional probability density function of X_i given $X_j = x_j$, where X_j could be two dimensional as well. $c_{13;2}(\cdot,\cdot;x_2)$ stands for the copula density of (X_1,X_3) given $X_2 = x_2$.

Putting this all together, as done also in Czado (2013), we get

Definition 3.2 (Pair copula decomposition in three dimensions) A **pair copula decomposition** of an arbitrary three dimensional probability density is given as

$$
\begin{aligned}
f(x_1,x_2,x_3) = {} & c_{13;2}(F_{1|2}(x_1|x_2),F_{3|2}(x_3|x_2);x_2)c_{12}(F_1(x_1),F_2(x_2))\\
& c_{23}(F_2(x_2),F_3(x_3))f_1(x_1)f_2(x_2)f_3(x_3)\ .
\end{aligned}
$$

So the joint three dimensional density can be expressed in terms of bivariate copulae and conditional distribution functions. One also sees that this decomposition is not unique, as we can condition also on the first or third variable (and adjust the bivariate copulae).

In Definition 3.2 our copula construction depends on the value of the second variable as $c_{13;2}(\cdot,\cdot;x_2)$ depends on x_2. In the rest of the thesis we will simplify this with the following assumption. For details on this, see Section 5.4 in Czado (2013) on page 66 et seqq.

Definition 3.3 (Simplifying assumption in three dimensions). Let X_2 be a random variable and $c_{13;2}(\cdot,\cdot;x_2)$ be a conditional copula density. Then the **simplifying assumption** is satisfied in three dimensions when the following holds

$$c_{13;2}(\cdot,\cdot;x_2) = c_{13;2}(\cdot,\cdot) \quad \text{for all } x_2 \ .$$

To construct a three dimensional copula, we actually do the pair copula decomposition "the other way round". We will specify arbitrary bivariate copula families for $c_{13;2}, c_{12}$ and c_{23} with corresponding parameters $\theta_{13;2}, \theta_{12}$ and θ_{23}. The parameters can be two dimensional. Then a valid parametric joint density is given by the following definition.

Definition 3.4 (Pair copula construction of a joint density in three dimensions). A three dimensional probability density with parameter vector $\boldsymbol{\theta} = (\theta_{13;2}, \theta_{12}, \theta_{23})$ can be constructed as follows

$$\begin{aligned} f(x_1,x_2,x_3) = {} & c_{13;2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2); x_2) c_{12}(F_1(x_1), F_2(x_2)) \\ & c_{23}(F_2(x_2), F_3(x_3)) f_1(x_1) f_2(x_2) f_3(x_3) \ , \end{aligned}$$

where $c_{13;2}(\cdot,\cdot;\theta_{13;2}), c_{12}(\cdot,\cdot;\theta_{12})$ and $c_{23}(\cdot,\cdot;\theta_{23})$ are arbitrary bivariate copula densities with specified parameters.

The whole implementation of visualizing trivariate copulae with the statistical software *R* is based on this definition and is covered in the following chapter.

Chapter 4

Visualization of Trivariate Copulae Using R

In this chapter we show how we realized the visualization of trivariate copulae using the statistical software *R*. We will do this by describing the actual workflow. Details on the code can be found in the appendix and is packed to **copulaSG**. Hereafter we show scenarios constructed first solely out of elliptical copulae as building copulae, followed by lots of other scenarios. At the end we apply our visualization to a practical example. In total we visualize 50 scenarios. A list of all scenarios is given in Appendix D.2 .

4.1 Workflow

0. **Load Relevant Packages**
   First we need *R* to know copulae. The *R* package **VineCopula** provides all functionalities for calculating with them. For visualization of three dimensional objects we use the packages **misc3d** and **rgl**. The package **xtable** allows us to export tables to the `.tex` format. We use these documents for presenting the specifications of scenarios in a standardized way.

1. **Function for Constructing R-Vine Matrix**
   The *R* package **VineCopula** calculates with copulae via an *RVineMatrix* object. For comfortably setting different copulae up, we define the function `RVMconstruction`.

2. **Function for Calculating the Probability Density**
   To calculate the probability density of a trivariate copula on a Z scale, we mainly implement Sklar's theorem and use *RVinePDF*.

3. **Functions for Plotting Contour Lines**
   We also provide two functions for plotting contour lines. The first one takes the calculated probability densities and plots the scenario from one point of view. The second one calls the first one to get the plot and afterwards takes snapshots of this plot from different points of view. These snapshots are consecutively numbered and saved in an appropriate folder. It is also possible to rotate the visualization manually.

4. **Functions for Handling the Scenarios**
   Now we set up a data frame for storing the different scenarios. We add all scenarios to this data frame and will then run through it to produce all plots with view a single line of code.

   As we want our scenarios to be as comparable as possible, we want to use the same τ values for most scenarios. In the *R* package **VineCopula** there already exists the function *BiCopTau2Par* for calculating parameter values. These are needed for setting up the *RVineMatrix* to calculate with. So we have to provide an extended version of this function, we name it *BiCopTau2ParX*, which is able to handle also copulae with two parameters.

   We further define ways to add scenarios to the data frame. For one we have to specify τ values and maybe a second parameter, and it calculates the first parameter (via *BiCopTau2ParX*), and the other calculates the τ value for given parameters.

   Then we define the function *scenarioToLatex*, which exports the specification of each scenario as a table to a `.tex` file.

   The last function to define is for handling all scenarios. The function *scenarioToFile* takes the number of the scenario and the data frame. It specifies the folder for saving the results of the visualization and calls all the other functions. Finally the folder will contain the snapshots as `.png` files, the configuration as a `.tex` file and a `.txt` document whose title shows what scenario is contained in this folder.

5. **Include Scenarios**
   Now we add a total number of 49 scenarios to our data frame.

6. **Visualize the Scenarios**
   The last thing to be done is running via the data frame and calling *scenarioToFile* for each scenario.

4.2 Setting for Visualizations

Before we actually start with the showing scenarios, we describe the settings for the visualizations in an extra section.

The visualization always contains four perspectives of one scenario and is done on the Z scale to provide comparability of different scenarios. We put each three dimensional contour plot in a box with **axes from** -3 **to** 3 (analogue to the two dimensional contour plots in Chapter 2) and add a cross to highlight the point of origin. The perspectives are chosen in a way that the contours are visible from every side. The plot in the upper left hand corner is the view from the front with the z_1-z_2 plane spanning in the back. We rotate this slowly clockwise round the z_2 axis to get the picture in the upper right hand corner. Then we continue rotating it clockwise round the z_2 axis until the z_2-z_3 plane spans in the back. This gives the perspective for the lower left hand picture. The last picture of the scenario, in the lower right hand corner, is taken from above.

Most scenarios are based on our **standard setting of $\boldsymbol{\tau}$ values** with $\tau_{12} = 0.8$, $\tau_{23} = 0.5$ and $\tau_{13;2} = 0.3$.

Furthermore we denote the first parameter of a copula for a building block by $\boldsymbol{\theta_1}$. So this refers to a ρ for elliptical copulae and to a θ of Archimedean copulae. The second parameter is denoted by $\boldsymbol{\theta_2}$ and refers to ν degrees of freedom of a Student t copula or to the δ for Archimedean copulae. If the copula family has no second parameter, like the Gaussian copula, we denote this by $\theta_2 = 0$. We introduce this notation in order to get a unique looking of the tables that describe the scenarios (here we will mix elliptical and Archimedean copulae).

4.3 First Visualizations

Now we are ready to give first examples of our visualization tool. We start with trivariate elliptical copulae, namely with trivariate Gaussian copulae and some trivariate t copulae.

4.3.1 Trivariate Gaussian Copulae

We start our visualization series with a trivariate Gaussian copula and our standard setting of τ values. The result is shown in Figure 4.1. One realizes the well known elliptical shape and strong dependence among the first and second variable, which can be seen best in the slim shape of contour lines in the upper left hand picture. Also one can imagine the zero tail dependence of Gaussian copulae resulting for example in radial symmetry. It is interesting that the shape of the 0.75 and 0.90 quantile can not be shown in a closed way. This can be interpreted as a lot of probability mass being distributed far away from the centre (compare this to the centralization of probability mass for a Frank copula, as shown in Figure 2.5).

This scenario is directly followed by actually quite the same scenario, but with rearranged τ values. Figure 4.2 shows the trivariate Gaussian copula with $\tau_{12} = 0.3$, $\tau_{23} = 0.5$ and $\tau_{13;2} = 0.8$. This figure shows the weak dependence among the first and second variable, which can be explored in the broad shape of contour lines in the upper left hand picture. The strong dependence of first and third variable can be seen in the slim shape of contour lines in the lower right picture. An interesting observation is that the 0.75 quantile is even less visualisable. It is easy to see in these plots, but one can also prove that the trivariate Gaussian copula satisfies the simplifying assumption Definition 3.3.

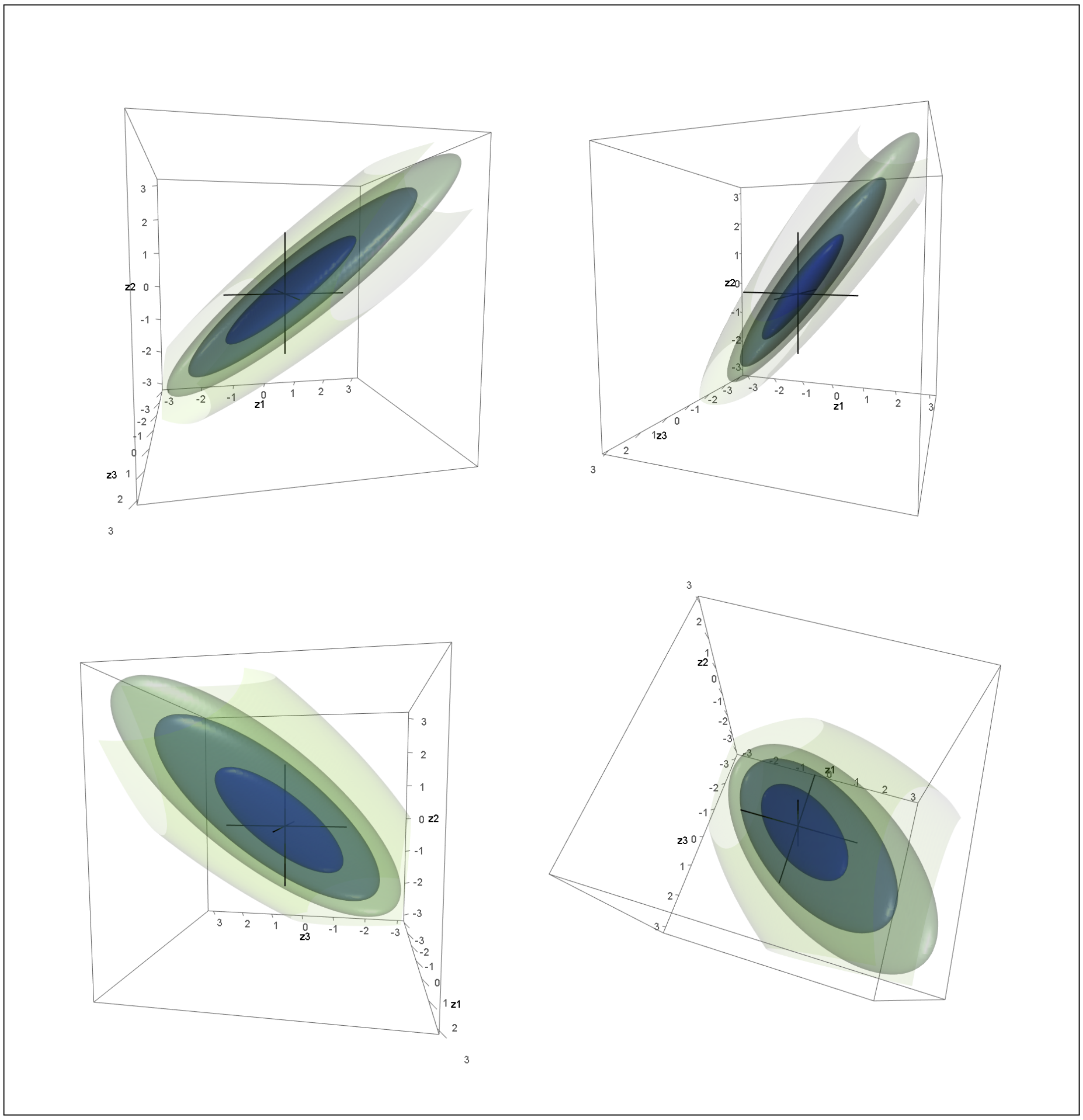

Figure 4.1:

Pair Copula	Family	τ	θ_1	θ_2
12	N	0.8	0.95	0
23	N	0.5	0.71	0
13|2	N	0.3	0.45	0

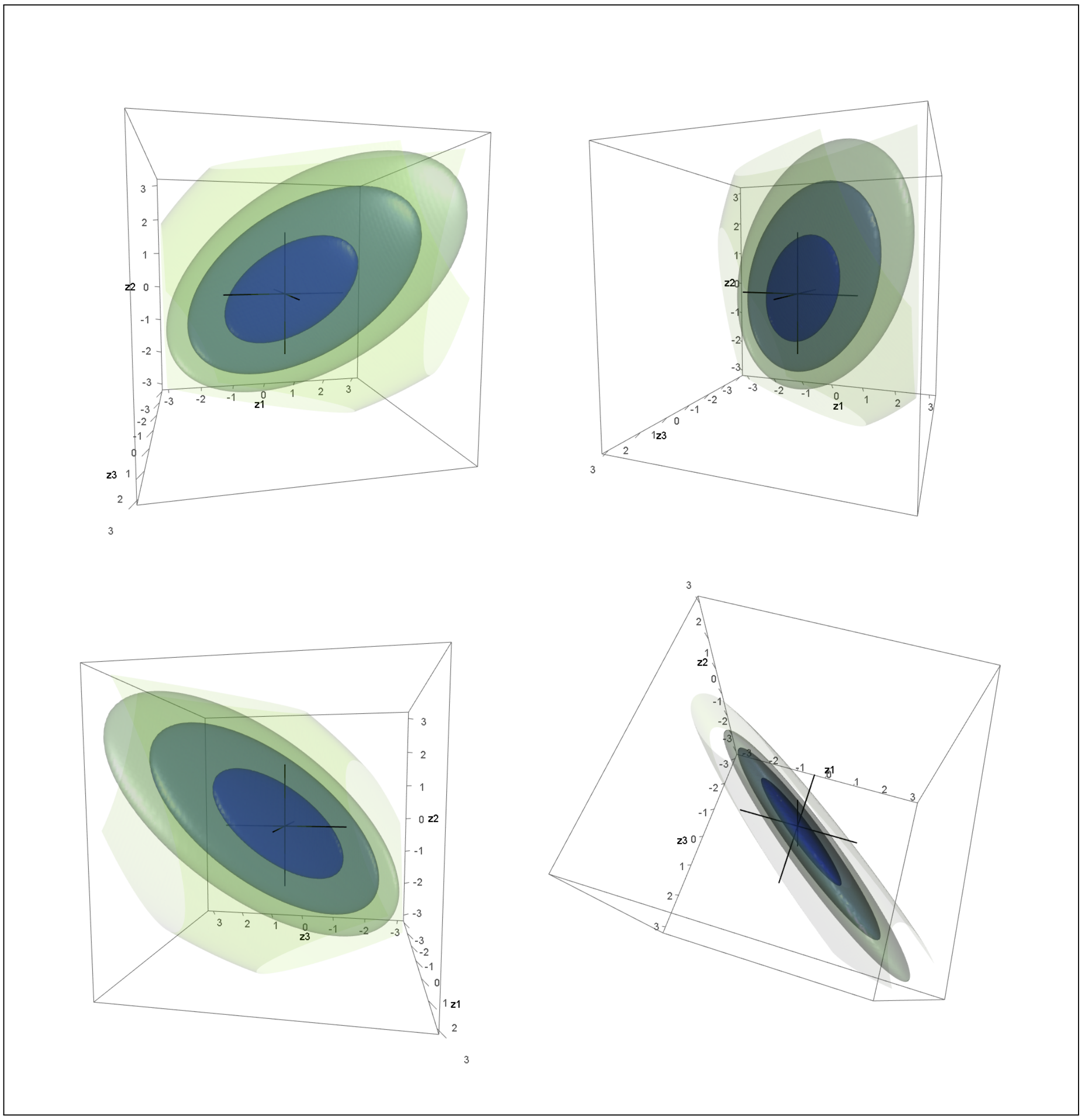

Figure 4.2:

Pair Copula	Family	τ	θ_1	θ_2
12	N	0.3	0.45	0
23	N	0.5	0.71	0
13|2	N	0.8	0.95	0

There are also two other ways of how to construct a Gaussian copula apart from setting τ values and calculating the parameters (which is our standard procedure). One can get to the pairwise construction directly from the correlation matrix R. We want to look at two special kinds of correlation matrices.

Definition 4.1 An **exchangeable correlation matrix** is given by

$$R = \begin{pmatrix} 1 & \rho & \rho \\ \rho & 1 & \rho \\ \rho & \rho & 1 \end{pmatrix} ,$$

with correlation coefficient $\rho \in [-1, 1]$. This correlation matrix R is positive definite for $\rho \in (-\frac{1}{2}, 1)$ and may be a reasonable choice when all pairs are expected to have the same dependence.

The **autoregressive model AR(1)** has a correlation matrix

$$R = \begin{pmatrix} 1 & \rho & \rho^2 \\ \rho & 1 & \rho \\ \rho^2 & \rho & 1 \end{pmatrix} ,$$

with $\rho \in [-1, 1]$. It may be used for modelling situations where variables are most dependent when they are near to each other.

A connection between the correlation matrix R and the pair copula construction with its dependency parameters $\theta_{13;2}$, θ_{12} and θ_{23} , as we described it in Definition 3.4 , is given in Section 2.6 in Aas et al. (2009) on page 8 et seq. Note that for a trivariate Gaussian copula, the θ's are only one dimensional. Thus we get the following connection among correlation matrix and parameters of a pair copula construction.

Theorem 4.2 Let $\theta_{13;2}$, θ_{12} and θ_{23} be the parameters of copulae $c_{13;2}, c_{12}$ and c_{23}. Furthermore let the correlation matrix R be given in the following way

$$R = \begin{pmatrix} 1 & \rho_{12} & \rho_{13} \\ \rho_{12} & 1 & \rho_{23} \\ \rho_{13} & \rho_{23} & 1 \end{pmatrix} .$$

Then the connection between both ways of stating correlation is given by

$$\begin{aligned} \rho_{12} &= \theta_{12} \\ \rho_{23} &= \theta_{23} \\ \rho_{13} &= \theta_{13;2}\sqrt{1-\theta_{12}^2}\sqrt{1-\theta_{23}^2} + \theta_{12}\theta_{23} \ . \end{aligned}$$

Proof. A proof can be found in Example 2.5 in Czado (2013) on page 24 et seq. □

As in our case the correlation matrix is given and we need the parameters of the pair copula construction, we derive the following connections.

Par	General	Exch.	AR(1)
θ_{12}	ρ_{12}	ρ	ρ
θ_{23}	ρ_{23}	ρ	ρ
$\theta_{13;2}$	$\frac{\rho_{13}-\rho_{12}\rho_{23}}{\sqrt{1-\rho_{12}^2}\sqrt{1-\rho_{23}^2}}$	$\frac{\rho(1-\rho)}{1-\rho^2}$	0

Figure 4.3 visualizes trivariate exchangeable Gaussian copulae with parameter ρ adjusted in a way that τ_{12} is 0.8 in the top row, 0.5 in the middle one and 0.3 in the bottom row. Remaining correlation parameters are determined due to the exchangeable structure. Figure 4.4 is produced in the same manner but with an underlying AR(1) structure.

The resulting visualizations show the impact of the characterizing parameter ρ. In the top its value is high (and so are θ_{12}, θ_{23} as well as τ_{12}, τ_{23}) and even the 0.95 quantile cannot be shown in closed shape. This high dependence results in long drawn-out elliptical forms.

The difference between an exchangeable and an AR(1) copula, which is the correlation value among the copula $c_{13;2}$, is hard to see in the visualizations. But by comparing the middle rows, one can realize that the 0.75 quantile is more closed for the AR(1) case, shown in Figure 4.4. If there is also dependence among this building block, as it is the case for the exchangeable copula, probability mass is fluctuating more out of the centre.

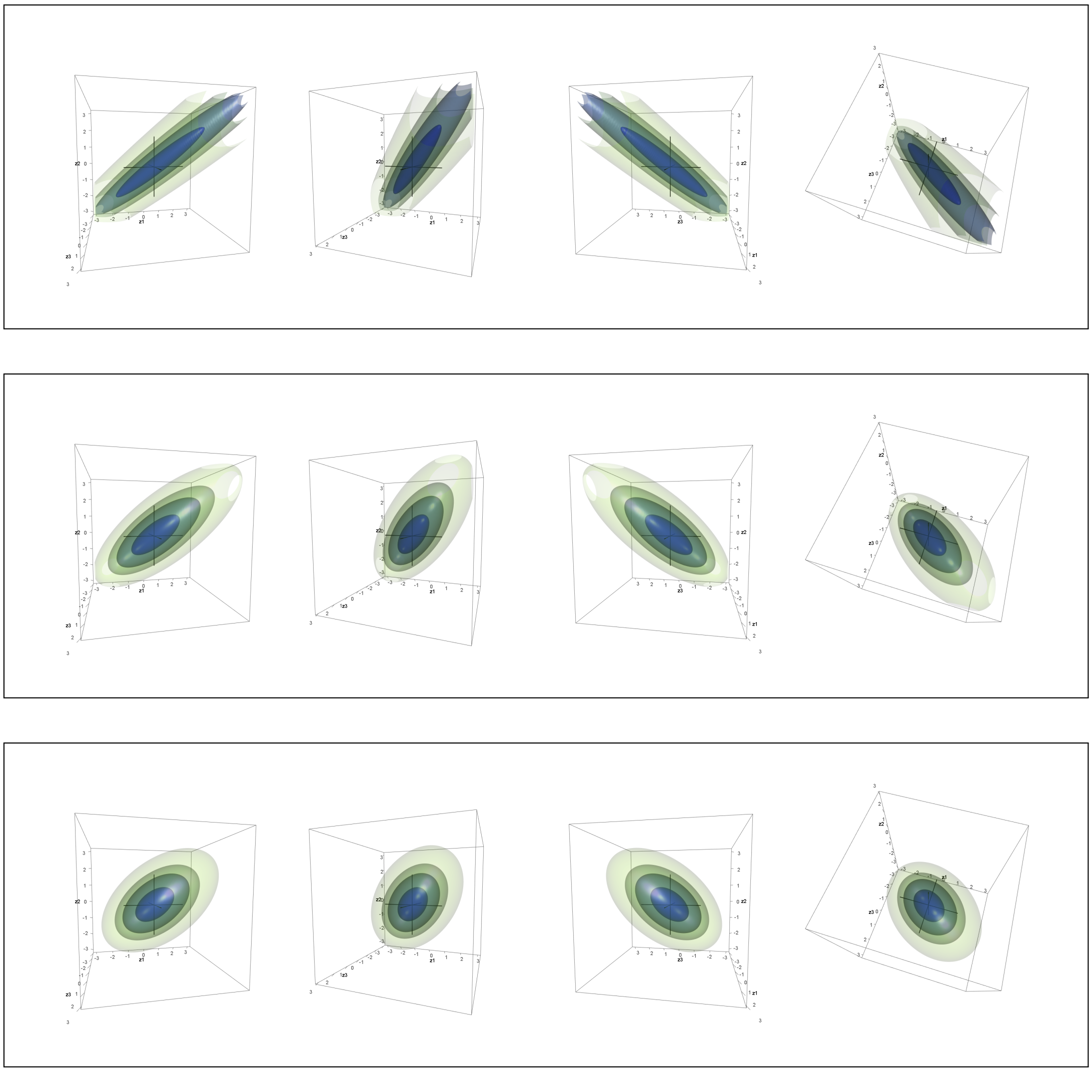

Figure 4.3: Exchangeable Gaussian copulae with:

	τ_{12}	θ_{12}	τ_{23}	θ_{23}	$\tau_{13;2}$	$\theta_{13;2}$
top	0.8	0.95	0.8	0.95	0.32	0.49
mid	0.5	0.71	0.5	0.71	0.27	0.42
low	0.3	0.45	0.3	0.45	0.20	0.31

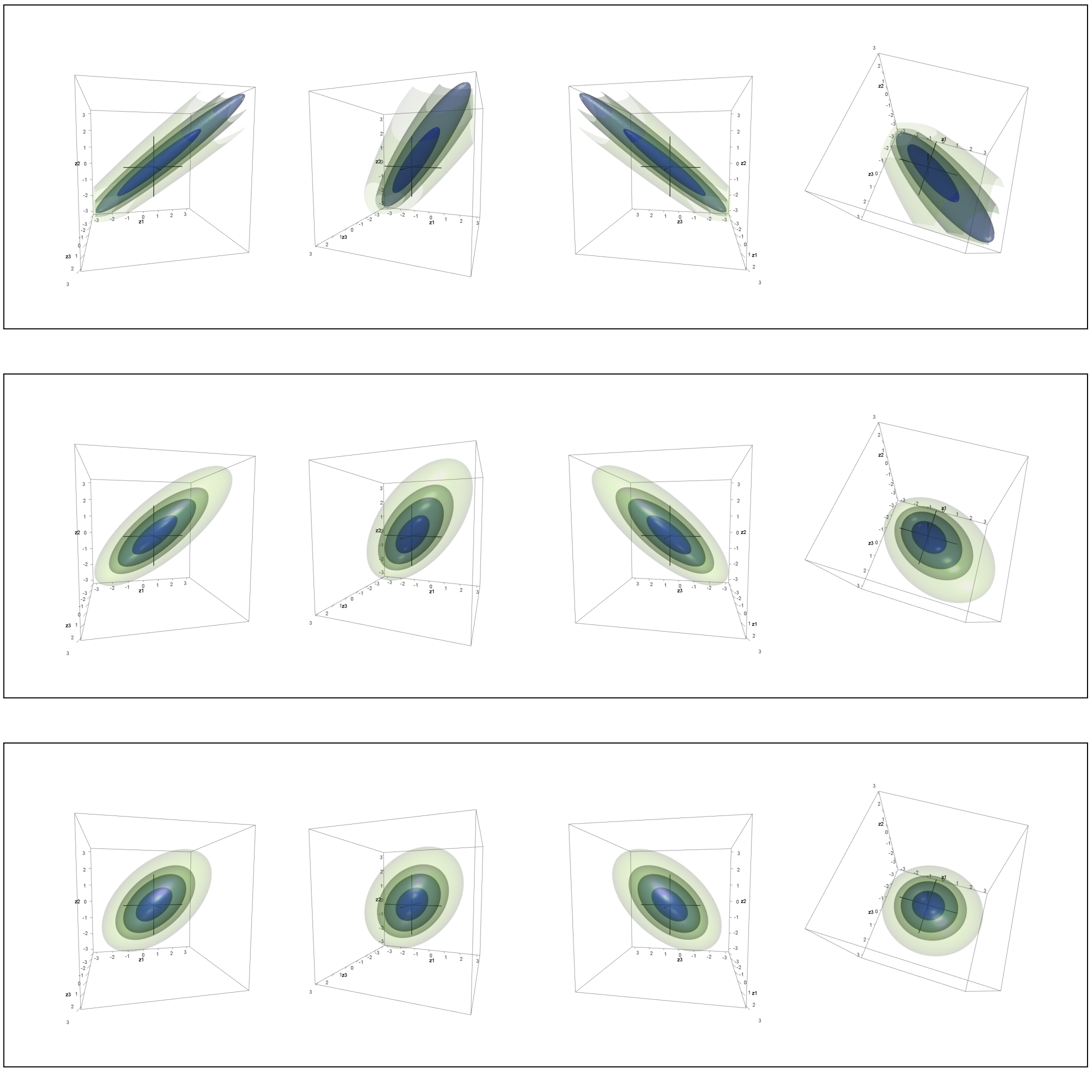

Figure 4.4: AR(1) Gaussian copulae with:

	τ_{12}	θ_{12}	τ_{23}	θ_{23}	$\tau_{13;2}$	$\theta_{13;2}$
top	0.8	0.95	0.8	0.95	0	0
mid	0.5	0.71	0.5	0.71	0	0
low	0.3	0.45	0.3	0.45	0	0

4.3.2 Trivariate t Copulae

Let us move on to the second member of elliptical copula family, the Student t copula. We introduced the bivariate t copula in Example 2.10 . Now we want to visualize trivariate Student t copula and show the influence of the degrees of freedom. Note that the degrees of freedom, encapsulated in ν and θ_2 , of the copula $c_{13;2}$ have to be increased by one, as one degree of freedom is lost due to the conditioning on the second variable.

We start with the isolated effect of degrees of freedom, visualized in Figure 4.5 . These are followed by the scenarios for our standard dependence setting in Figure 4.6. We bring both cases on one double-page, so they are directly comparable.

A trivariate t copula with zero τ values and few degrees of freedom, as visualized in the top row of Figure 4.5, looks like a cube. It is always symmetric and with increasing degrees of freedom, it takes the shape of a ball, which is the shape of perfect independence. Note that τ values of a Student t copula do not depend on the degrees of freedom.

With more dependence among the variables, as given in Figure 4.6, the cube gets more torn apart. One also sees the well known fact, that a Student t distribution behaves more and more like a Gaussian distribution with increasing degrees of freedoms. This is visible by comparing the last row with Figure 4.1.

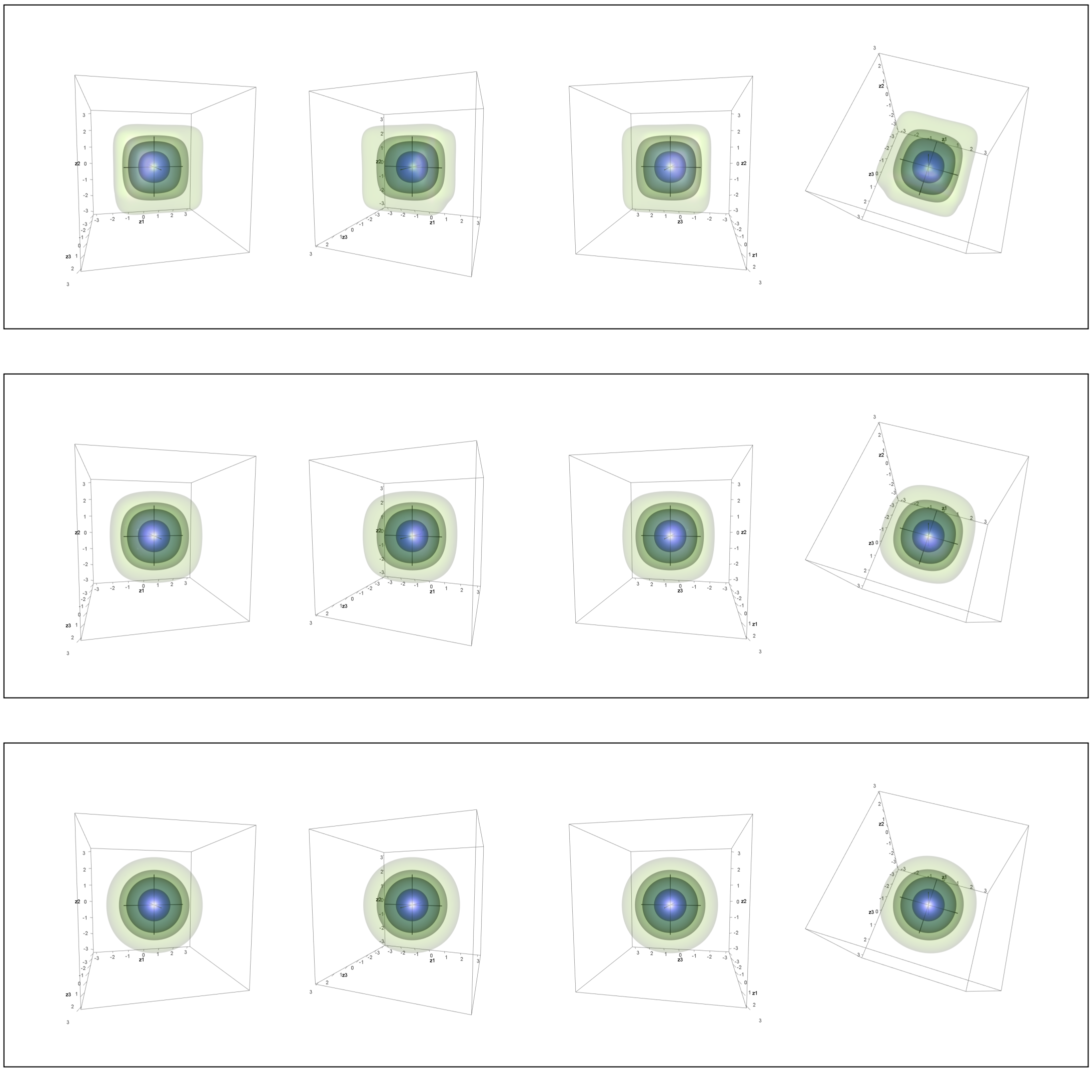

Figure 4.5: Student t copulae with:

	Copula c_{12}				Copula c_{23}				Copula $c_{13;2}$			
	Fam	τ	θ_1	θ_2	Fam	τ	θ_1	θ_2	Fam	τ	θ_1	θ_2
top	t	0	0	3	t	0	0	3	t	0	0	4
mid	t	0	0	6	t	0	0	6	t	0	0	7
low	t	0	0	25	t	0	0	25	t	0	0	26

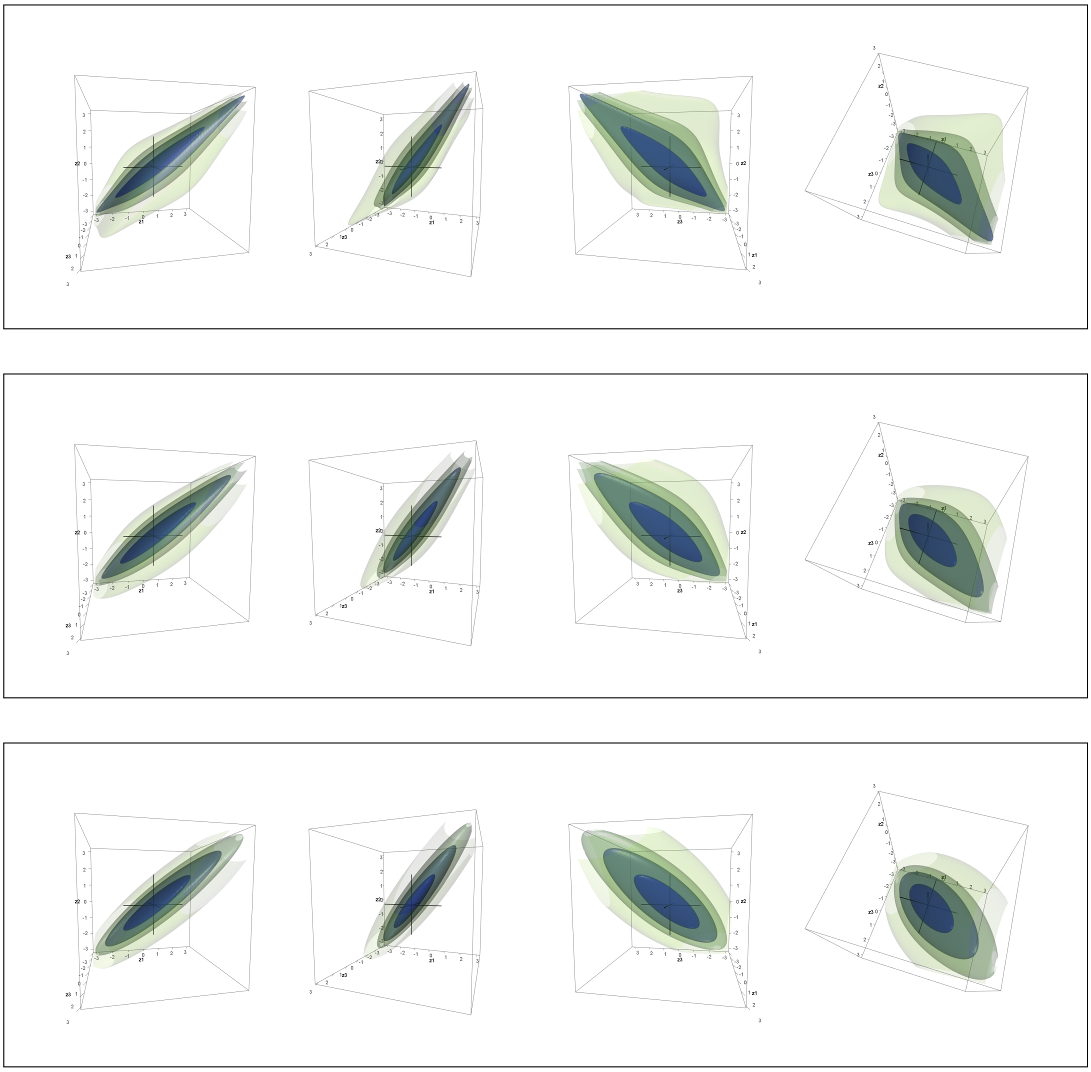

Figure 4.6: Student t copulae with:

	Copula c_{12}				Copula c_{23}				Copula $c_{13;2}$			
	Fam	τ	θ_1	θ_2	Fam	τ	θ_1	θ_2	Fam	τ	θ_1	θ_2
top	t	0.8	0.95	3	t	0.5	0.71	3	t	0.3	0.45	4
mid	t	0.8	0.95	6	t	0.5	0.71	6	t	0.3	0.45	7
low	t	0.8	0.95	25	t	0.5	0.71	25	t	0.3	0.45	26

4.3.3 Independence Copula

Next we want to study a special case of copulae. In a situation with no dependence among the variables, we get an independent copula.

Definition 4.3 A trivariate **independence copula** is given by

$$\begin{aligned} c(u_1, u_2, u_3) &= 1 \\ C(u_1, u_2, u_3) &= u_1 u_2 u_3 \,. \end{aligned}$$

The contours of an independence copula are perfectly symmetric balls as there is no dependence among the variables. Therefore Kendall's τ of an independence copula is equal to 0. But it is not true that for $\tau = 0$ the corresponding contour lines are also perfectly symmetric balls. We have observed this already for the Student t copula in Figure 4.5 and now want to visualize this fact in the following figure.

The independence copula is shown in the top row of Figure 4.7 . The contour lines are perfectly symmetric balls as mentioned above.
In the mid row we visualize the trivariate Gaussian copula built out of bivariate Gaussian copulae with $\tau = 0$ as building blocks. These look like perfect balls like for the independence copula.
In the last row we used a Gaussian, t and Clayton copula as building blocks and had to deal with the fact that a Clayton copula can not be used for $\tau = 0$. Thus we assigned a very low value of 0.01 to it. The result is quite near to a ball, but the influence of the three degrees of freedom of the t copula can be seen, especially in the quadratic shape of contour lines in the third picture.

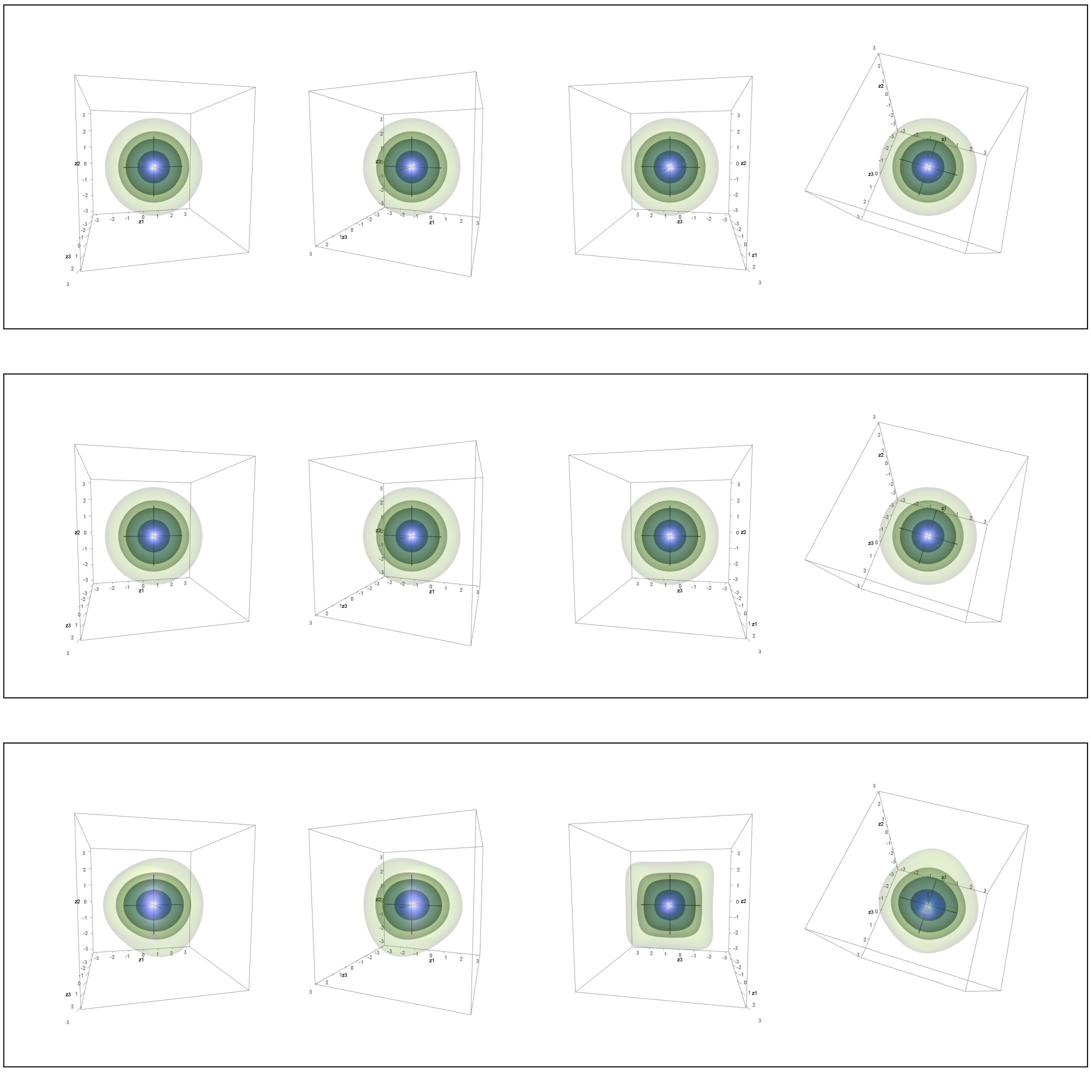

Figure 4.7: Independence copulae with:

	Copula c_{12}				Copula c_{23}				Copula $c_{13;2}$			
	Fam	τ	θ_1	θ_2	Fam	τ	θ_1	θ_2	Fam	τ	θ_1	θ_2
top	I	0	0	0	I	0	0	0	I	0	0	0
mid	N	0	0	0	N	0	0	0	N	0	0	0
low	N	0	0	0	t	0	0	3	C	0.01	0.02	0

4.4 Further Visualizations

Now we are ready for a lot more visualizations of various scenarios. In the following we will stick to visualizing one scenario on a single page as we did for our first visualizations of trivariate Gaussian copulae in Figure 4.1 and Figure 4.2 and mainly use the setting described in Section 4.2 .

4.4.1 Scenarios out of Archimedean Copulae with one Parameter

We move on to four Archimedean copulae with one parameter, so $\theta_2 = 0$. First we visualize trivariate copulae solely built out of either Clayton, Gumbel, Frank or Joe copulae as building blocks. After that we visualize trivariate copulae built out of a combination of these bivariate copula families.

Figure 4.8 visualizes the trivariate copula out of Clayton copulae and with our standard τ values. We recognize the high lower tail dependence of a Clayton copula, especially among the first and second variable, in each picture.

The high upper tail dependence of a Gumbel copula is visible in Figure 4.9 , which is based solely on Gumbel copulae. By taking a close look at the upper right picture this constellation seems to be curved a little to the top, which is rather an illusion than a characteristic of a Gumbel copula.

Frank copulae are the underlying building blocks for Figure 4.10 . The interesting high concentration of probability mass in the centre is again greatly visible in the upper left picture. Here one can see that contour lines are narrowed at the point of origin and are spread out at the tails.

We take a look on Joe copulae as building blocks in Figure 4.11 . The high upper tail dependence among the first and second variable, modelled by a Joe copula, can be seen perfectly in the upper pictures.

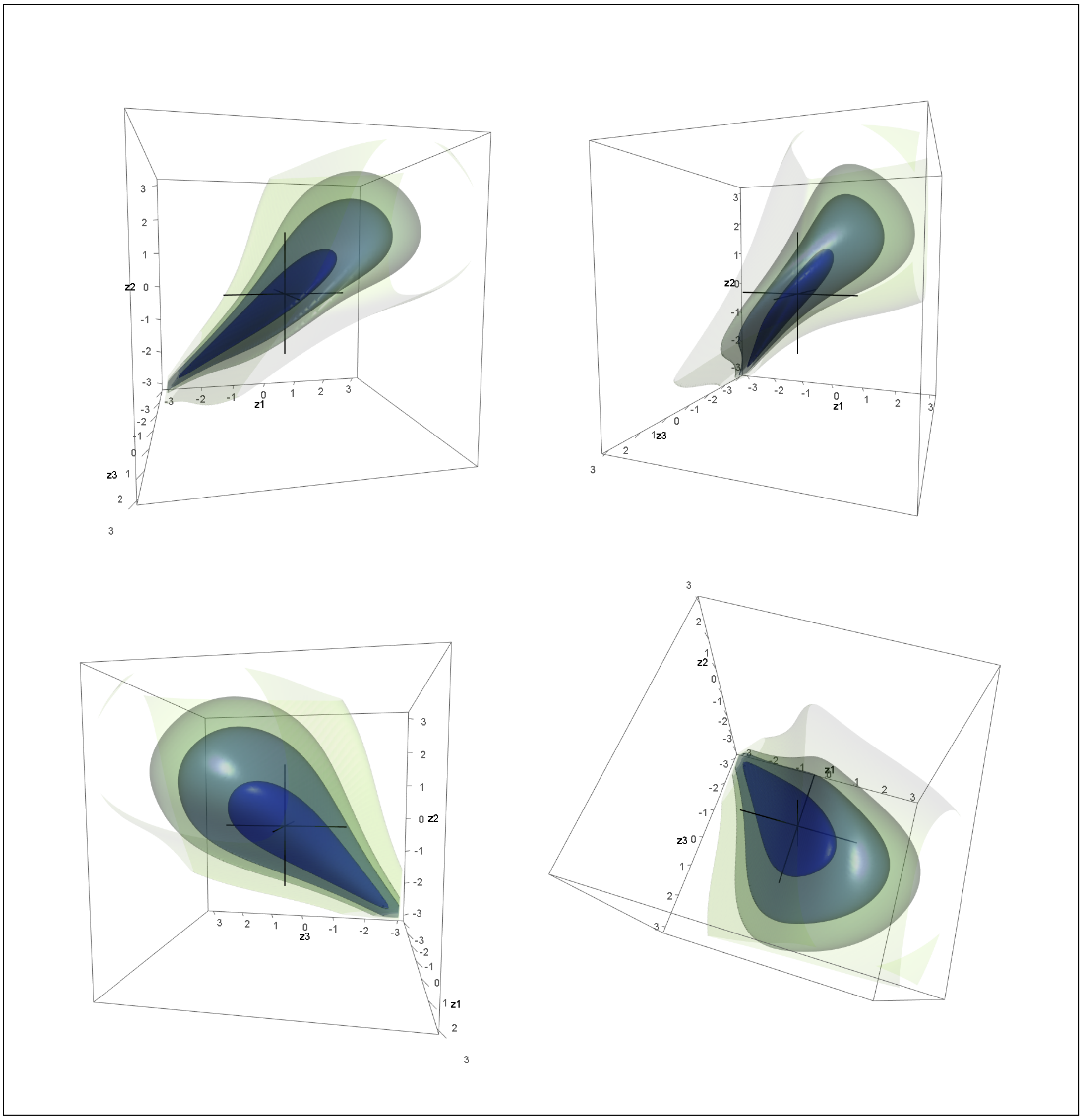

Figure 4.8:

Pair Copula	Family	τ	θ_1	θ_2
12	C	0.8	8	0
23	C	0.5	2	0
13|2	C	0.3	0.86	0

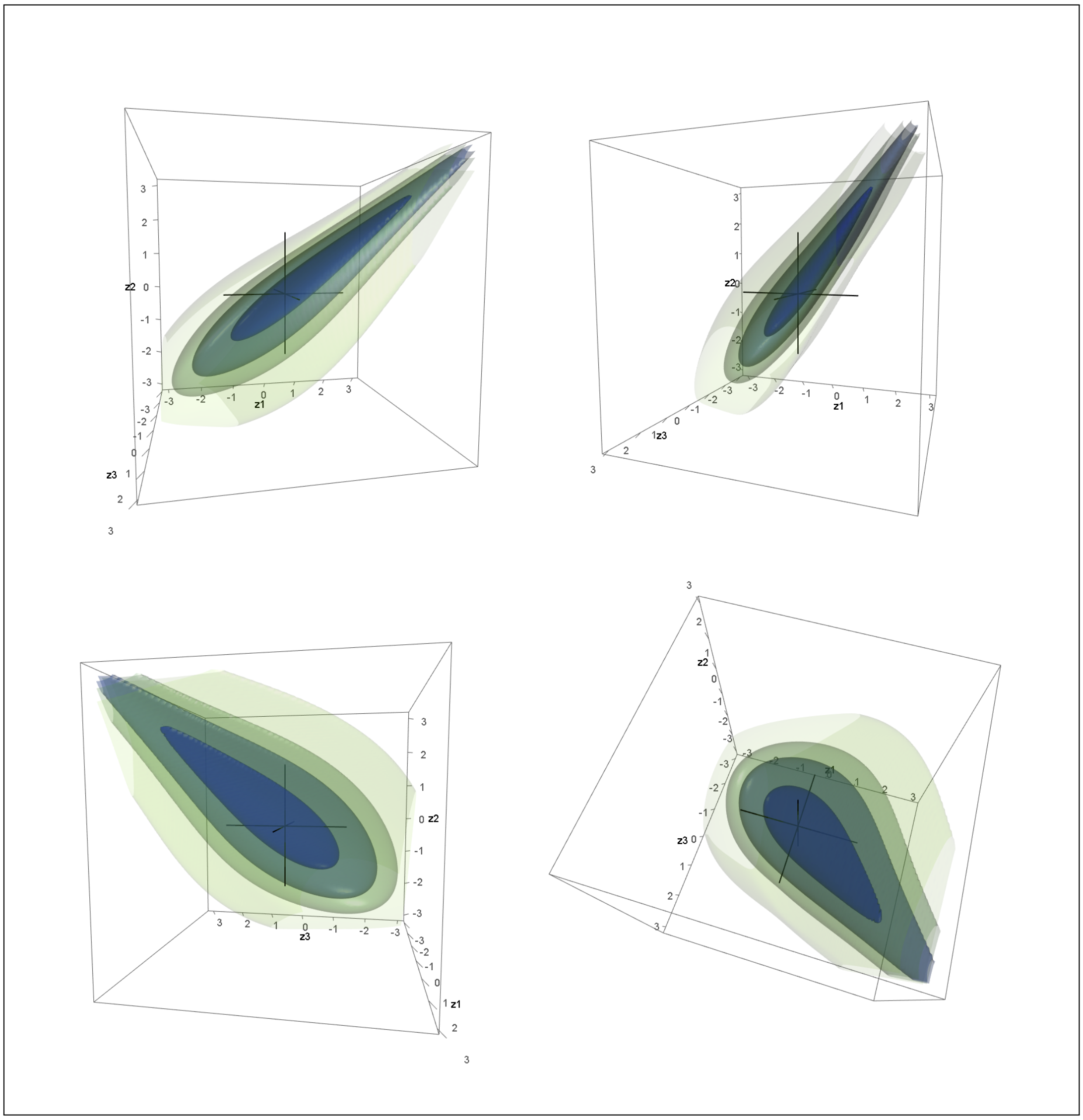

Figure 4.9:

Pair Copula	Family	τ	θ_1	θ_2
12	G	0.8	5	0
23	G	0.5	2	0
13|2	G	0.3	1.43	0

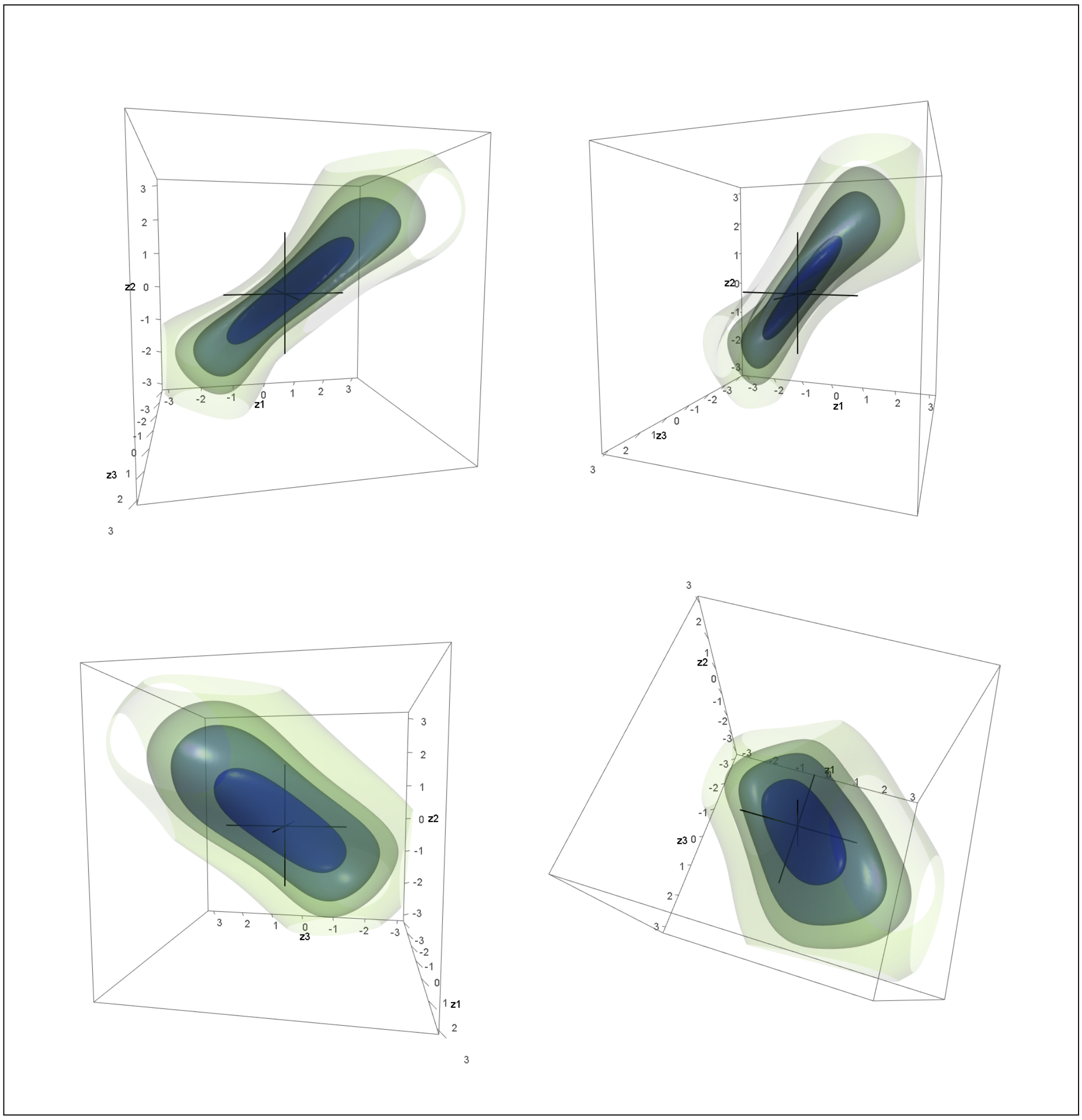

Figure 4.10:

Pair Copula	Family	τ	θ_1	θ_2
12	F	0.8	18.19	0
23	F	0.5	5.74	0
13|2	F	0.3	2.92	0

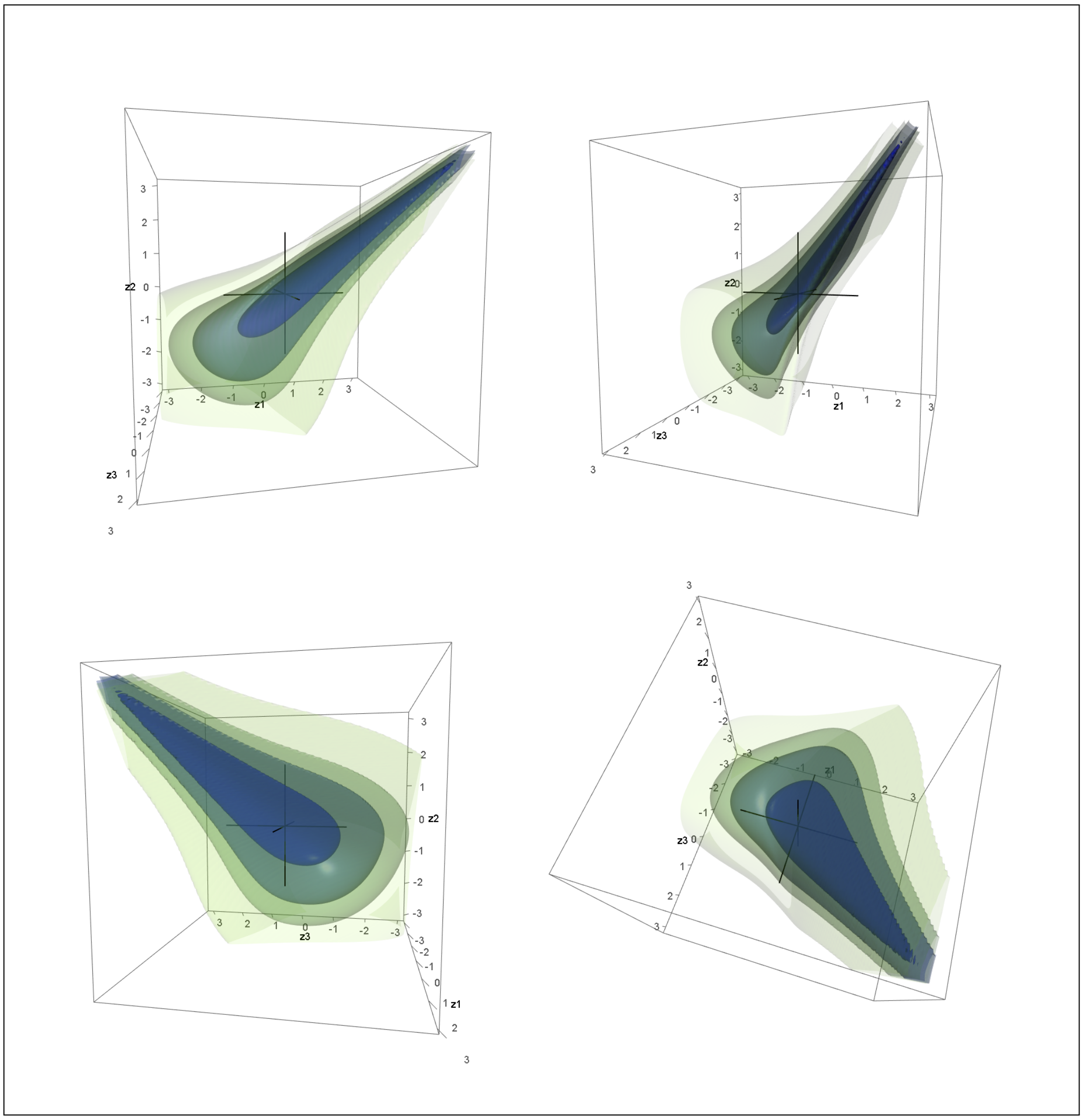

Figure 4.11:

Pair Copula	Family	τ	θ_1	θ_2
12	J	0.8	8.77	0
23	J	0.5	2.86	0
13|2	J	0.3	1.77	0

After this exploration of trivariate copulae solely based on either Clayton, Gumbel or Frank copulae, we now want to visualize trivariate copulae with building blocks mixed out of these copula families.

In Figure 4.12 we start with the order Clayton, Gumbel and Frank for the underlying copulae and our standard τ values. So a Clayton copula models the strong dependence ($\tau = 0.8$) among the first and second variable, a Gumbel copula the dependence with $\tau = 0.5$ of second and third variable and a Frank copula is used for the remaining pair copula $C_{13|2}$ with $\tau = 0.3$. The resulting pictures show the high lower tail dependence of a Clayton copula as can be seen especially in the upper right picture. The characteristics of a Frank copula is less visible due to the low dependence modelled by it.

Now we reverse the order of the building blocks and illustrate the resulting copula in Figure 4.13. So the Frank copula models high dependence among first and second variable with high concentration of probability mass at the centre. This is greatly visible in both upper pictures. The upper tail dependence of a Gumbel copula can be seen in the lower left hand picture.

Another scenario out of Archimedean copulae with one parameter can be found in Figure 4.14, which is build out of a Joe, Frank and Gumbel copula. So the upper tail dependence among first and second variable is modelled via a Joe copula and can be discovered graphically in the upper right picture.

Our final example for this section is given in Figure 4.15 and consists out of a Gumbel, Joe and Clayton copula. A Gumbel copula models again high upper tail dependence but also lower tail dependence. Thus the resulting shape is more tightened than for the scenario before.

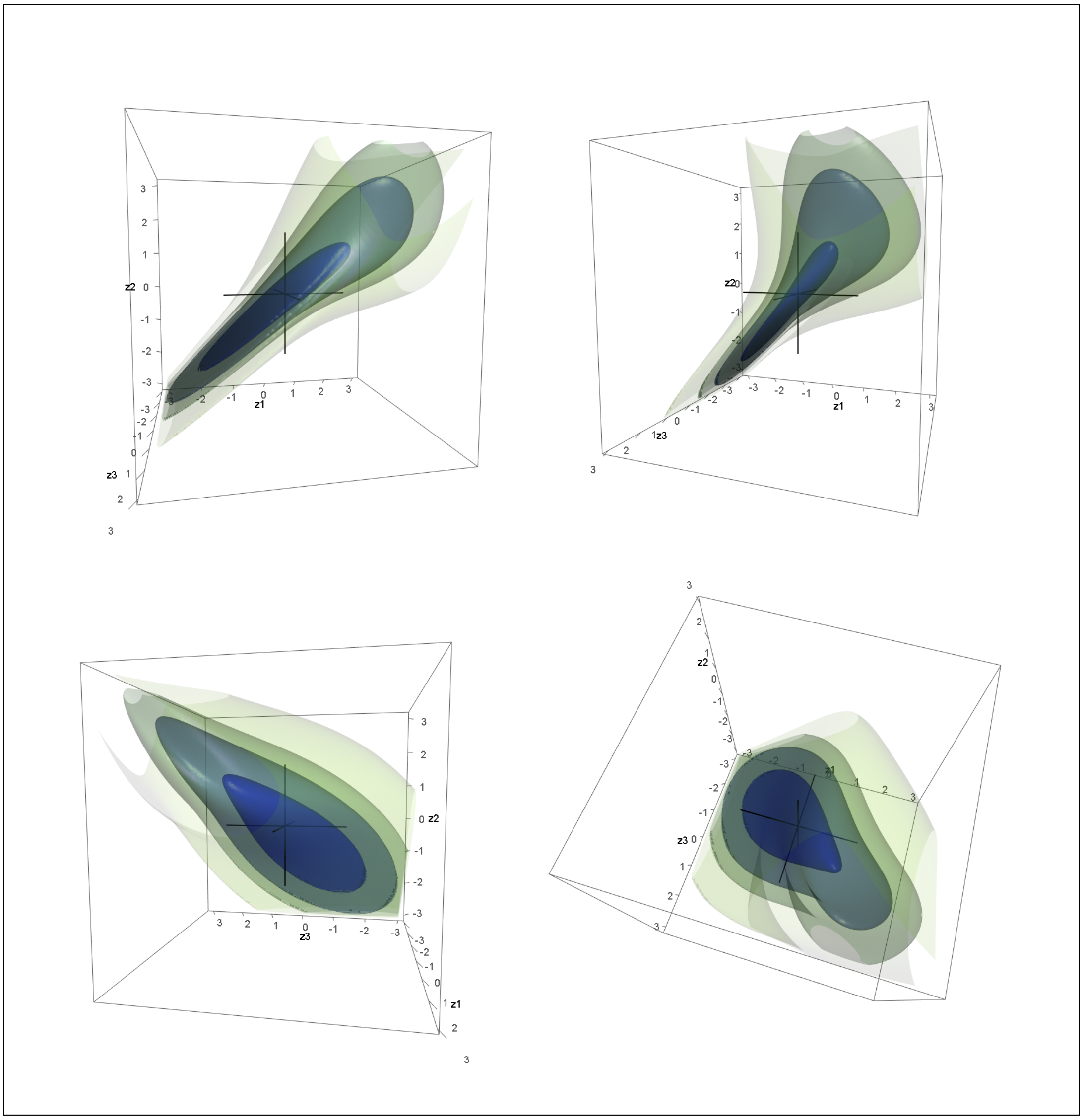

Figure 4.12:

Pair Copula	Family	τ	θ_1	θ_2
12	C	0.8	8	0
23	G	0.5	2	0
13|2	F	0.3	2.92	0

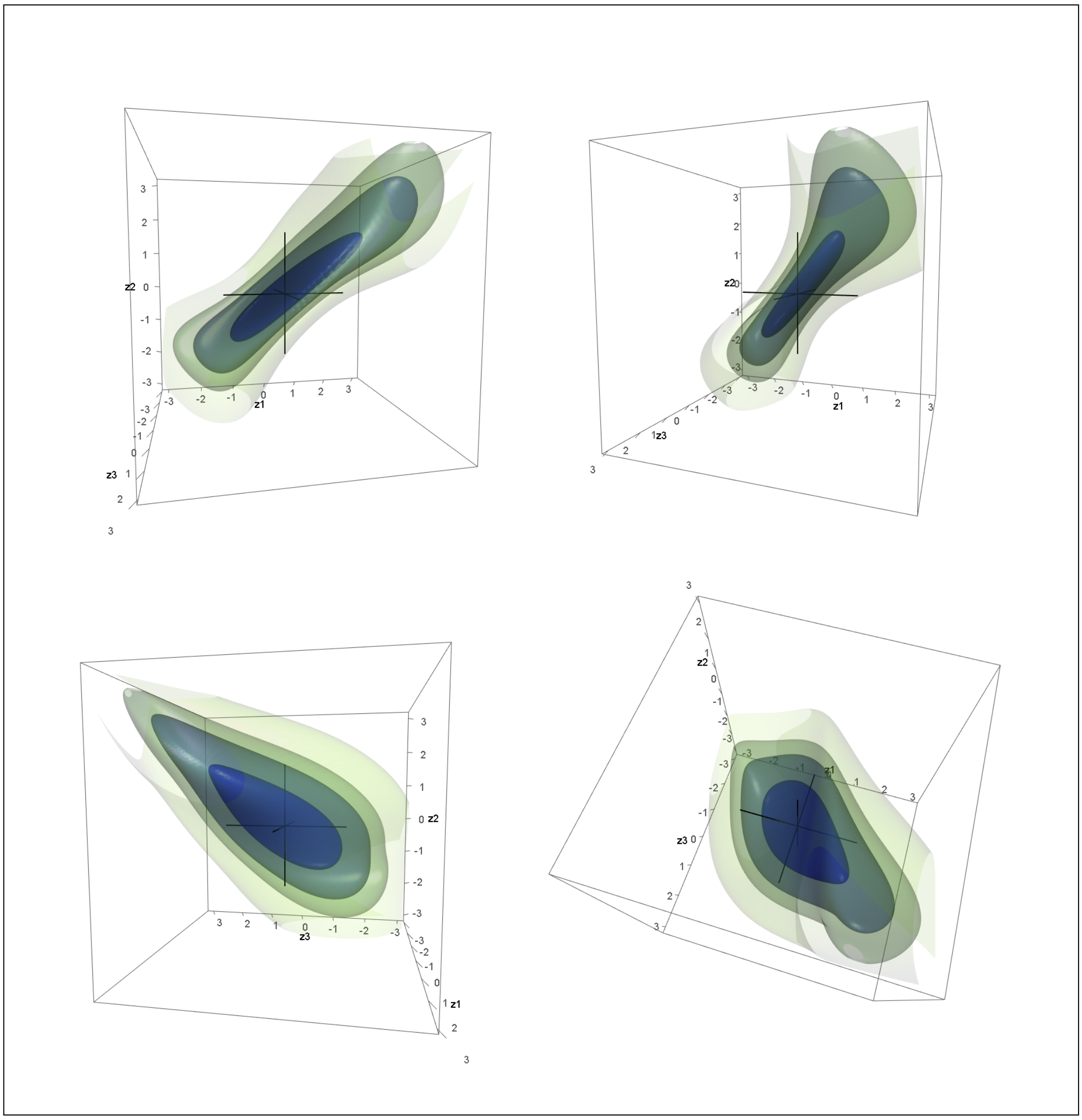

Figure 4.13:

Pair Copula	Family	τ	θ_1	θ_2
12	F	0.8	18.19	0
23	G	0.5	2	0
13|2	C	0.3	0.86	0

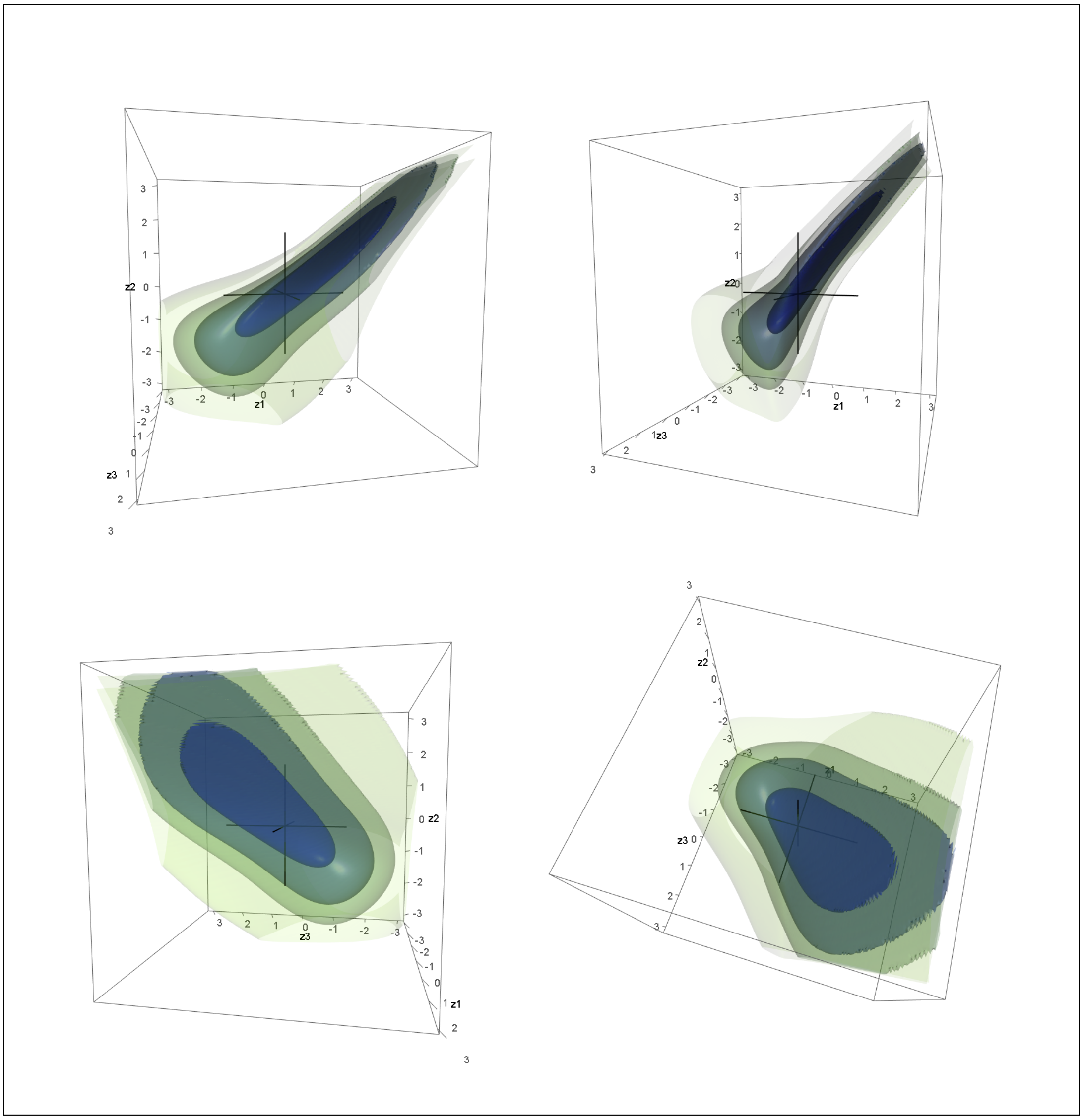

Figure 4.14:

Pair Copula	Family	τ	θ_1	θ_2
12	J	0.8	8.77	0
23	F	0.5	5.74	0
13|2	G	0.3	1.43	0

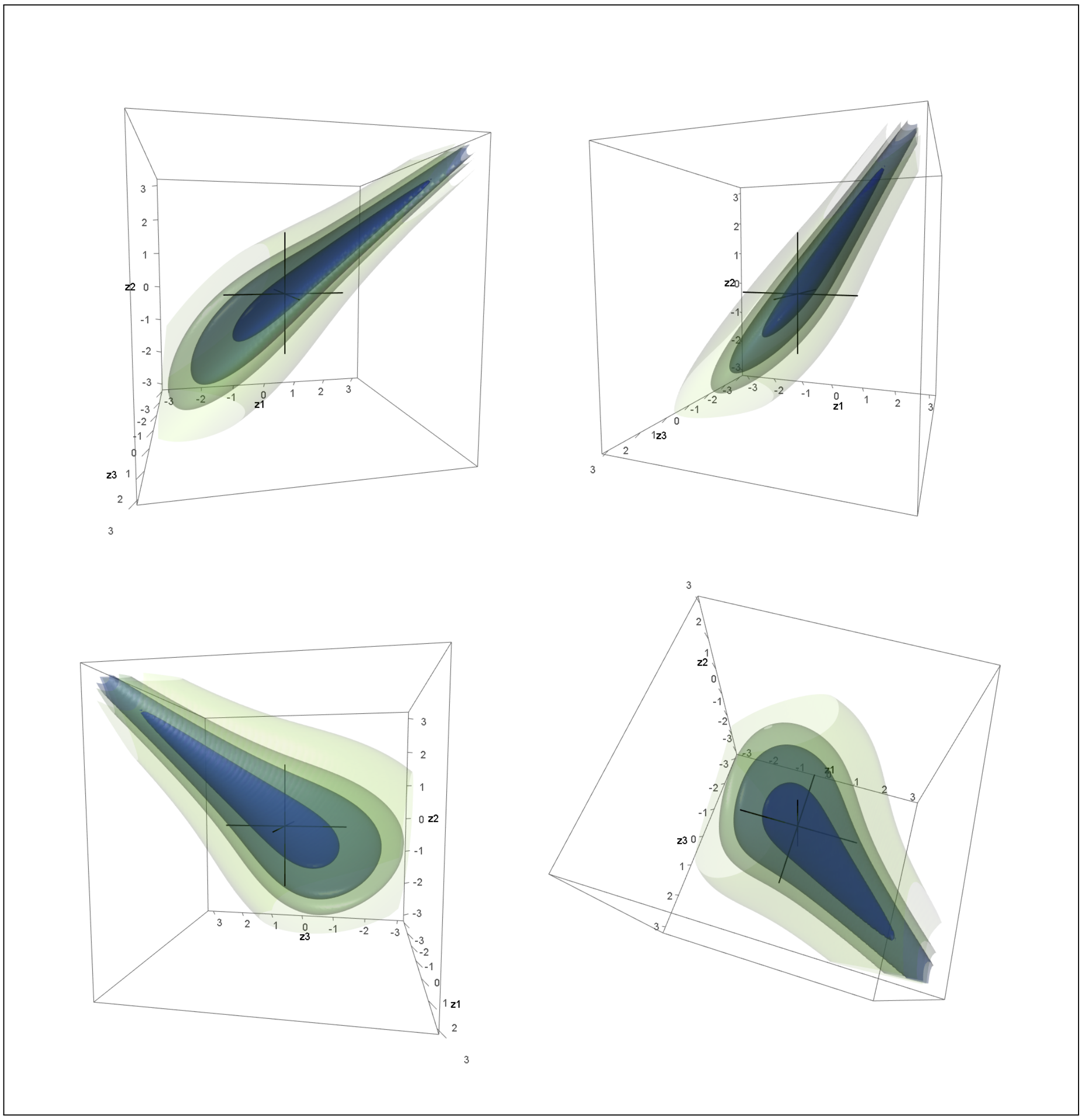

Figure 4.15:

Pair Copula	Family	τ	θ_1	θ_2
12	G	0.8	5	0
23	J	0.5	2.86	0
13|2	C	0.3	0.86	0

4.4.2 Scenarios out of Archimedean Copulae with two Parameters and Tawn Copulae

Next we get to Archimedean copulae with two parameters and Tawn copulae as building blocks. The *R* package **VineCopula** has implemented six scenarios and their rotated versions. Firstly we take a look on trivariate copulae solely out of either BB1, BB6, BB7, BB8, Tawn type 1 or Tawn type 2 as bivariate building blocks. Afterwards we mix these copula families together to get other interesting scenarios. Throughout we use our standard τ values of 0.8 , 0.5 and 0.3 .

Figure 4.16 visualizes the trivariate copula based on BB1 copulae as building blocks. This copula family is characterized by its high lower tail dependence, which is especially recognizable in the upper pictures.

The opposite, high upper tail dependence, is a feature of BB6 copulae, as used for Figure 4.17. The shape is similar to those of the Joe copulae in Figure 4.11.

Figure 4.18 shows the trivariate copula constructed out of BB7 copulae as building blocks. The upper tail dependence is also high in this case and especially visible in the upper pictures. It is interesting to see that even the contour line of the 0.99 quantile cannot be shown in closed shape. This can be observed in the lower left picture and is a sign for a concentration of probability mass towards this direction, so really high upper tail dependence. One could say, the dependences of all copulae (c_{12} , c_{23} and $c_{13|2}$) are reinforcing each other and sum up to this result.

The last BB family implemented in the *R* package **VineCopula** is used for Figure 4.19. A BB8 copula has tightened contour lines at the point of origin and thus some similarity with a Frank copula (especially for high values of the first parameter). This can also be underpinned by comparing this figure to Figure 4.10.

Now we turn over to the Tawn families. Figure 4.20 shows the trivariate copula built on Tawn type 1 copulae. For high dependence, as it is the case among the first and second variable, the characteristic bulge is greatly visible in the upper pictures. We say the bulge is located on the lower side, as z_1 values are low at its position. The position of the bulge of a trivariate Tawn type 1 copula is on the opposite site as it was for the bivariate Tawn type 1 copula (compare to Figure 2.11). This can be explained in the way that our implementation of trivariate copulae in a strict sense specifies c_{21} instead of c_{12}. This is usually not observed because of symmetric copula families.

On the other hand the bulge on the upper side characterizes the Tawn type 2 copula and can be seen best in the upper pictures of Figure 4.21 .

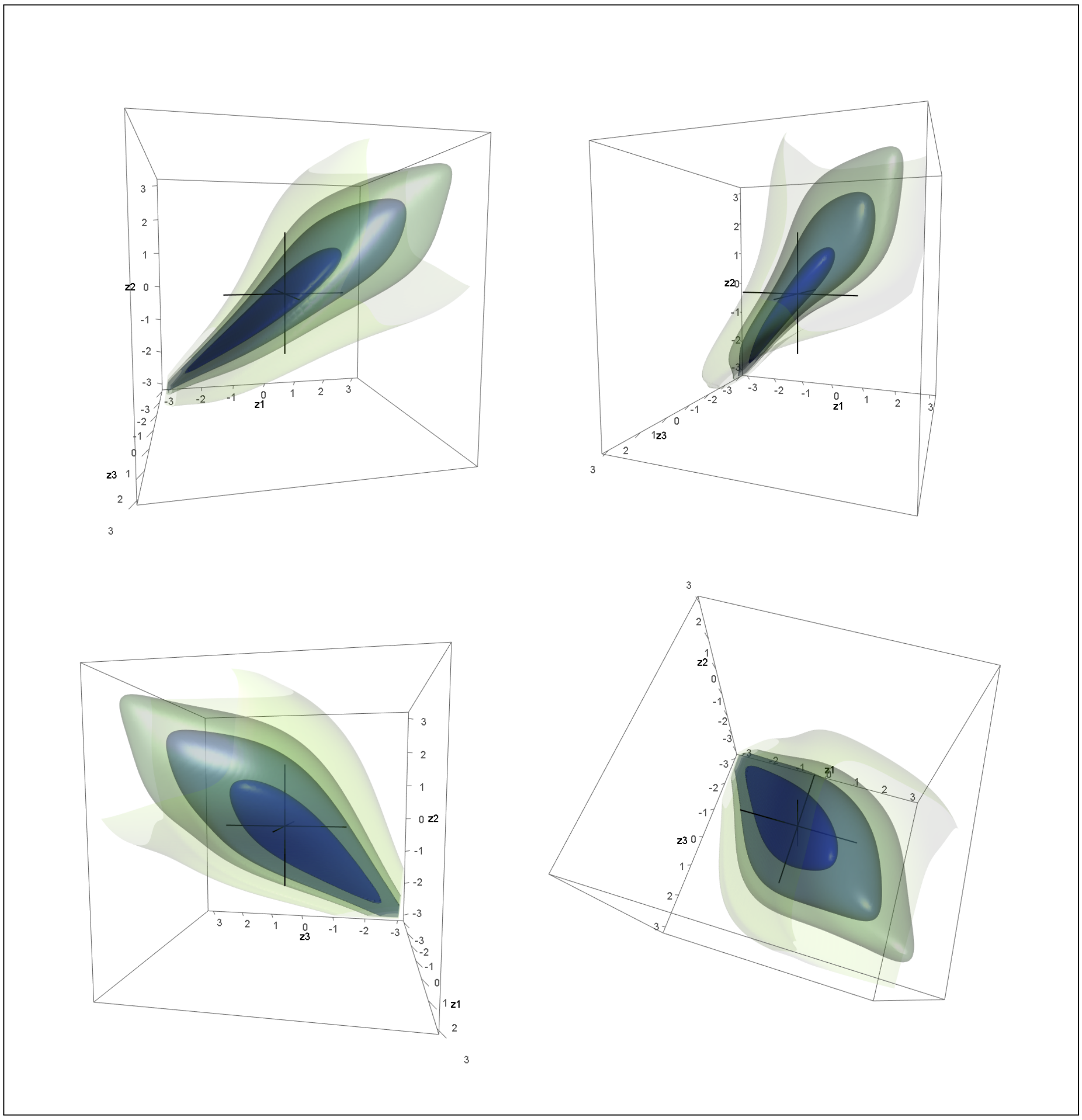

Figure 4.16:

Pair Copula	Family	τ	θ_1	θ_2
12	BB1	0.8	5.69	1.3
23	BB1	0.5	1.08	1.3
13|2	BB1	0.3	0.2	1.3

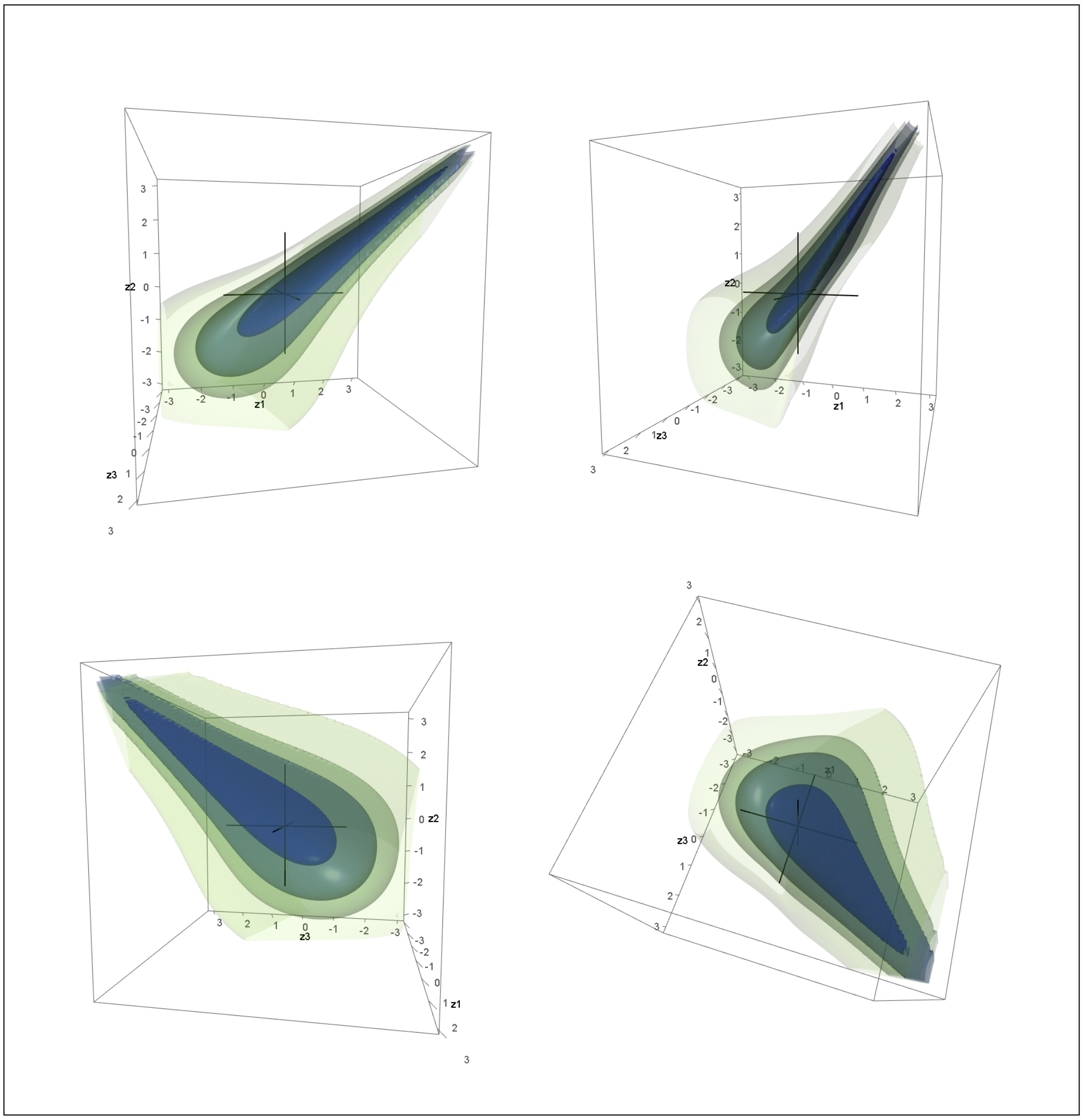

Figure 4.17:

Pair Copula	Family	τ	θ_1	θ_2
12	BB6	0.8	5.46	1.5
23	BB6	0.5	1.98	1.3
13|2	BB6	0.3	1.17	1.3

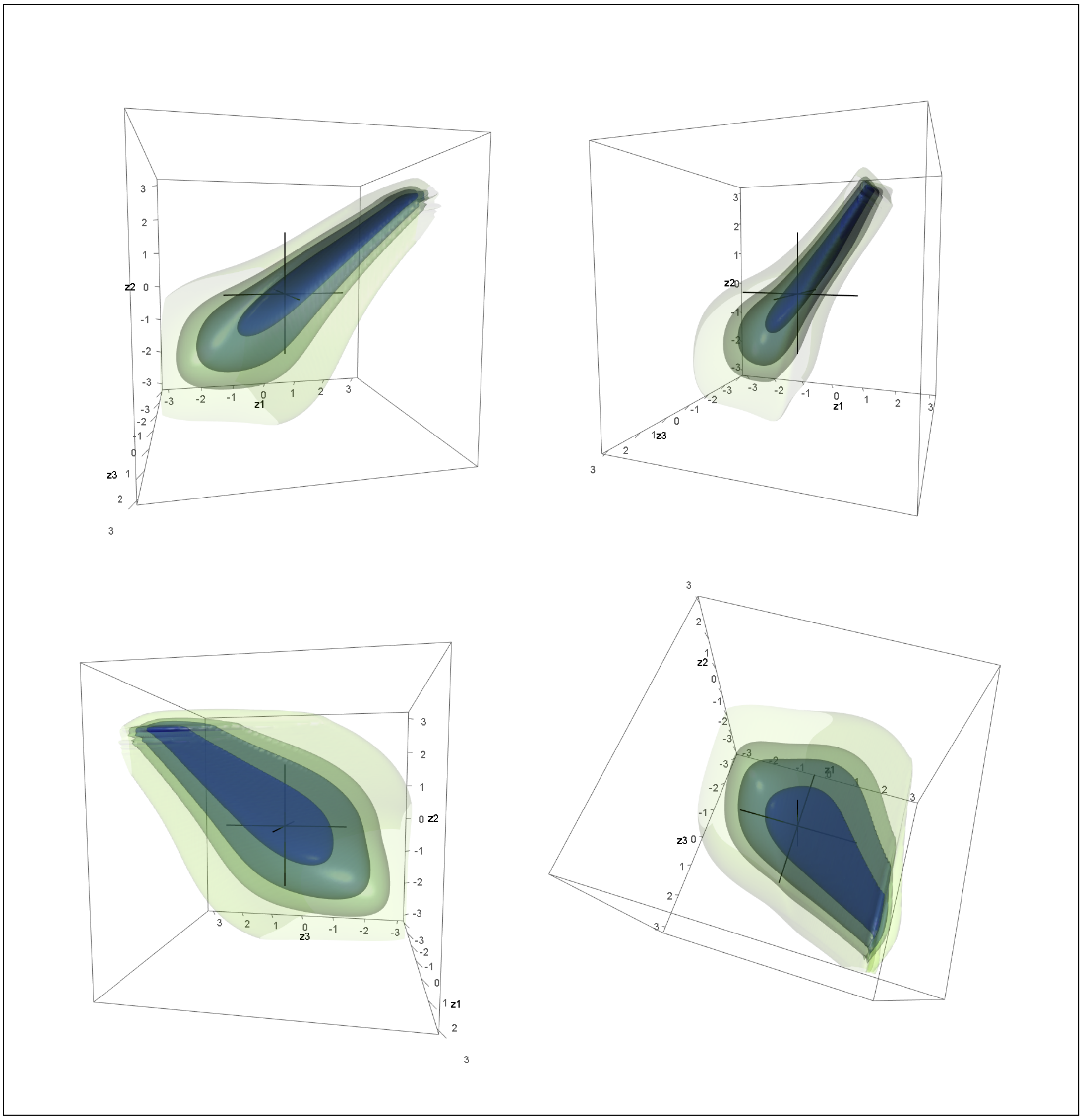

Figure 4.18:

Pair Copula	Family	τ	θ_1	θ_2
12	BB7	0.8	8.43	0.4
23	BB7	0.5	2.5	0.4
13|2	BB7	0.3	1.42	0.4

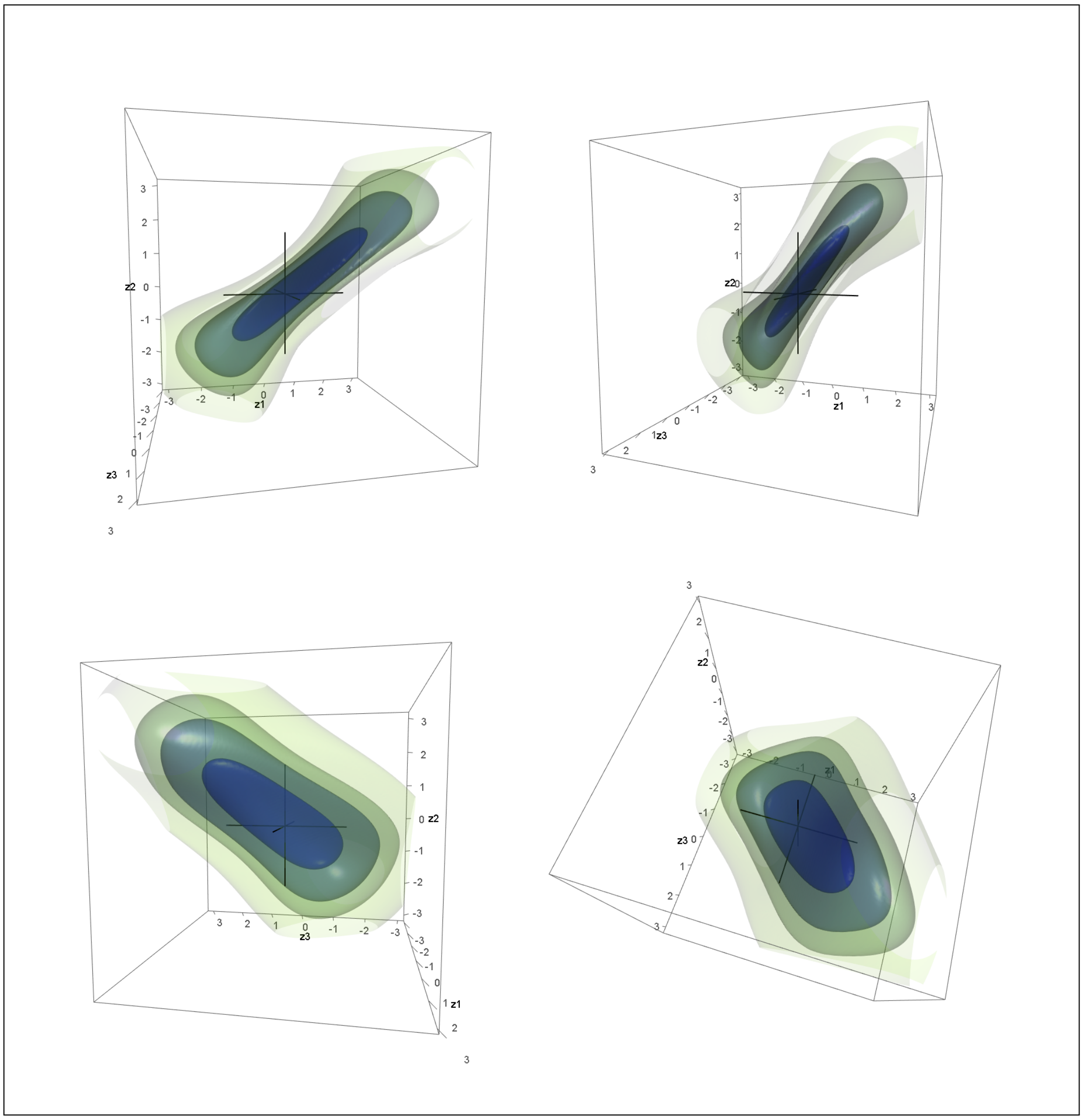

Figure 4.19:

Pair Copula	Family	τ	θ_1	θ_2
12	BB8	0.8	16.91	0.7
23	BB8	0.5	5.58	0.7
13|2	BB8	0.3	3.2	0.7

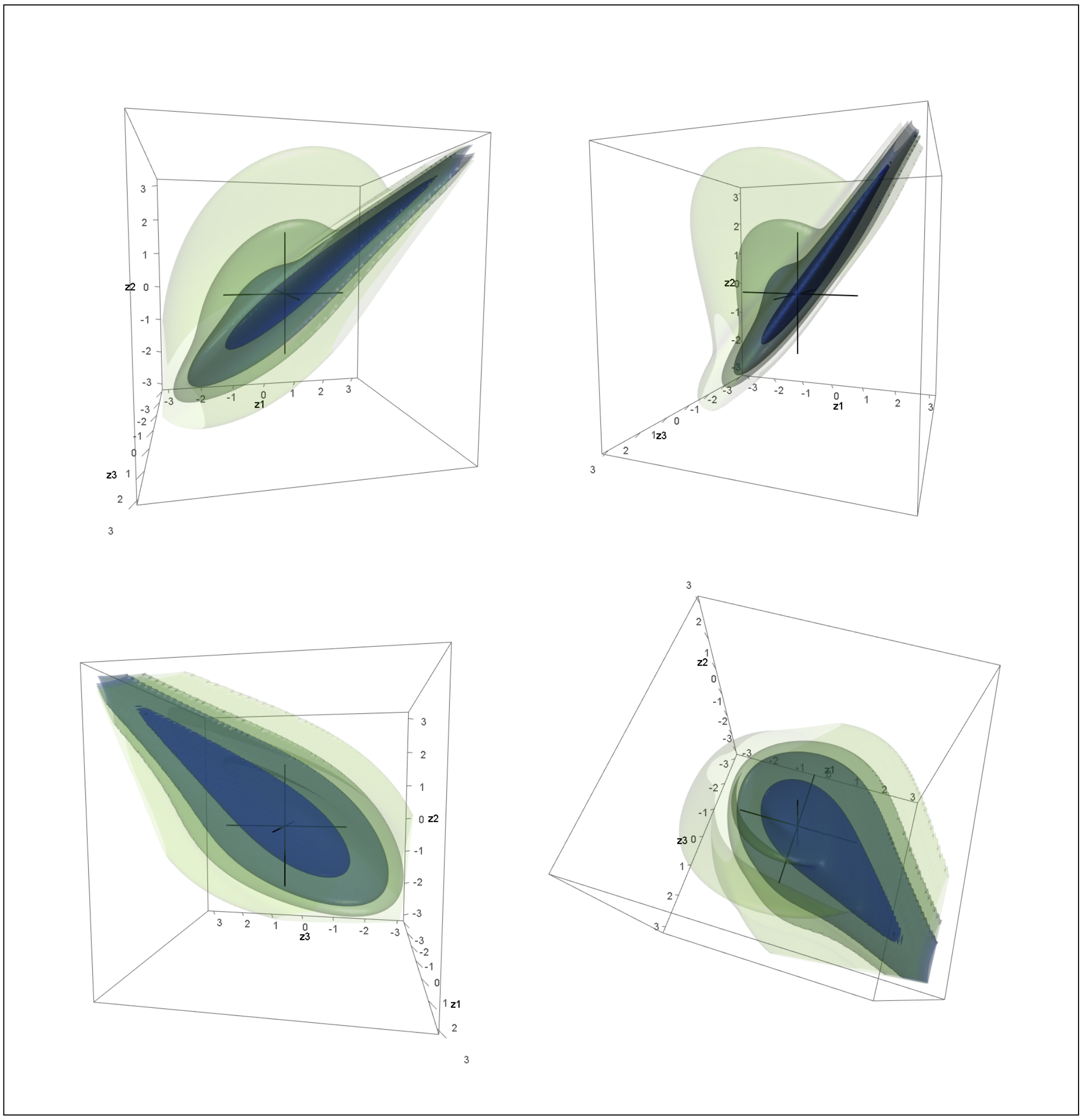

Figure 4.20:

Pair Copula	Family	τ	θ_1	θ_2
12	Tawn	0.8	8.28	0.9
23	Tawn	0.5	2.16	0.9
13|2	Tawn	0.3	1.47	0.9

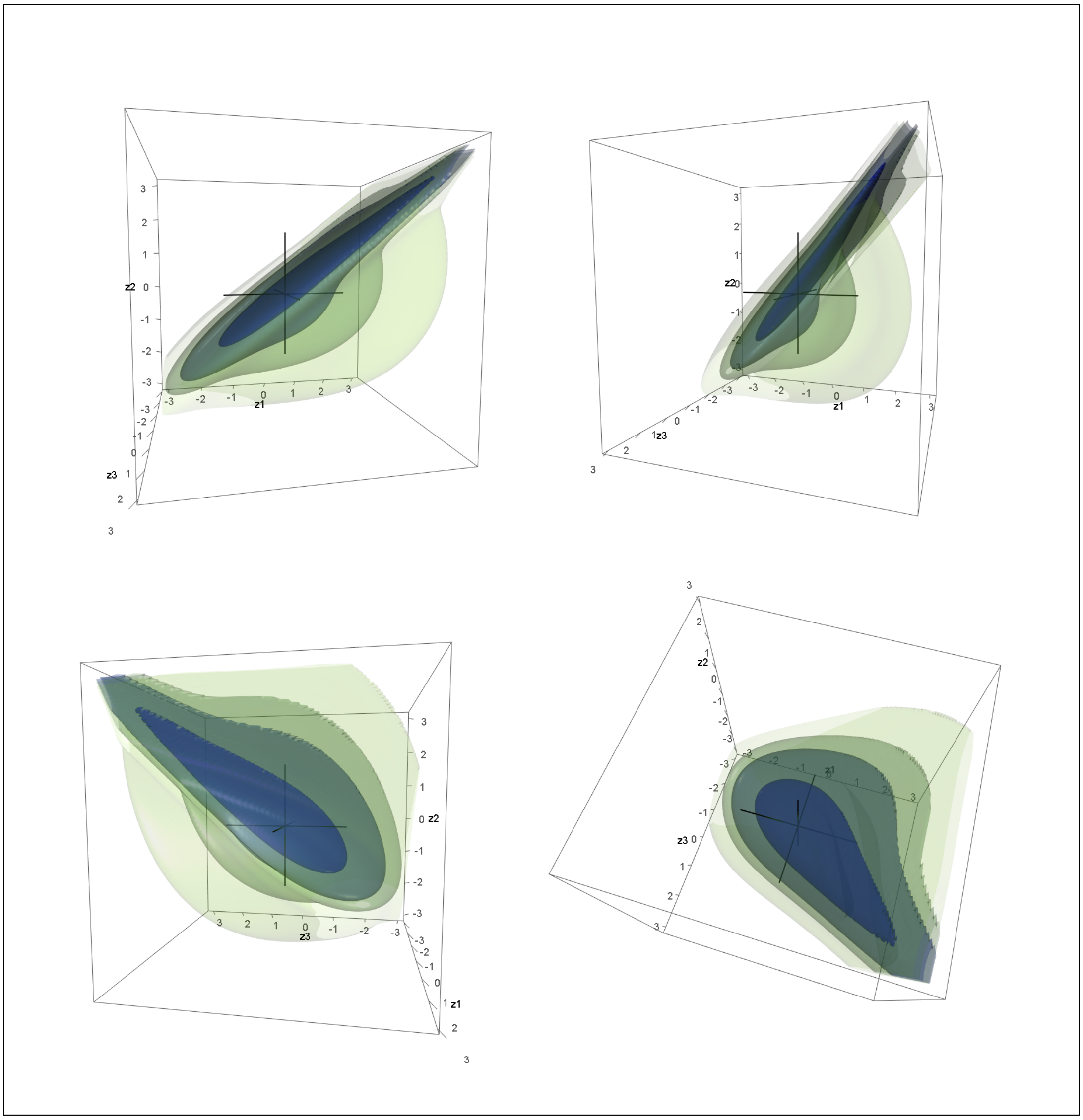

Figure 4.21:

Pair Copula	Family	τ	θ_1	θ_2
12	Tawn2	0.8	8.28	0.9
23	Tawn2	0.5	2.16	0.9
13|2	Tawn2	0.3	1.47	0.9

Now we go on and mix bivariate Archimedean copula families with two parameters together to construct trivariate copulae.

Figure 4.22 is built out of BB1, BB6 and BB7 copulae. Here the influence of the building blocks on different dependence structures among the variables is nicely visible. A BB1 copula models high lower tail dependence together with some upper tail dependence and this can be seen in the upper pictures. Just from these two pictures one wouldn't imagine high upper tail dependence and no lower tail dependence, as modelled by a BB6 copula, among the second and third variable. But exactly this gets visible in the lower left picture. So our perspectives are chosen appropriately.

The combination of a BB8, Tawn type 1 and Tawn type 2 copula result in the trivariate copula visualized in Figure 4.23. The high concentration of probability mass in the centre, which causes the similarity of the BB8 and Frank copulae, can be seen in the upper pictures. The characteristic bulges of Tawn copulae are invisible as the dependence they are modelling is low.

If dependence modelled by a Tawn copula gets higher, the bulge will get visible. This is the case in Figure 4.24. Here a Tawn type 1 copula models the high dependence among first and second variable and remaining building blocks consist out of a BB1 and BB8 copula. The bulge on the lower side (low values of z_1) is greatly visible in the upper pictures and the lower tail dependence of a BB1 copula can be recognized in the lower left hand picture. The lower right hand picture is interesting when keeping in mind that there is a bulge in this picture. We see that this is difficult to observe and so perspectives in the upper pictures are necessary to get an impression of the whole scene.

The last scenario of this section is visualized in Figure 4.25 and consists out of a Tawn type 2, BB6 and BB7 copula. The characteristic bulge on the upper side (high values of z_1) is visible in every picture except the one on the lower right hand side. But this picture shows us the (weak) upper tail dependence among the first and third variable modelled by a BB7 copula. The effect of upper tail dependence in this picture is enlarged by the upper tail dependence modelled by the other copulae.

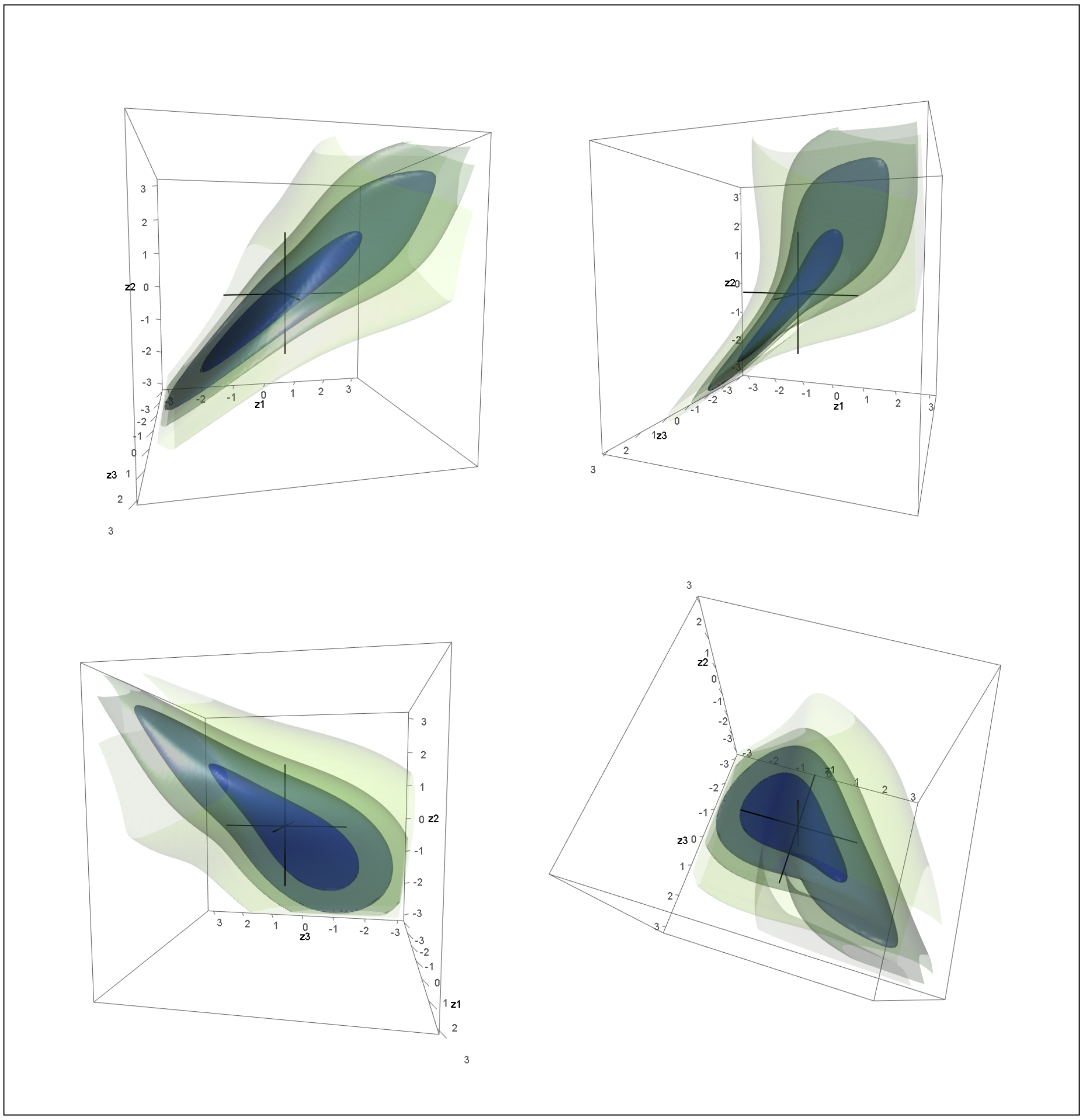

Figure 4.22:

Pair Copula	Family	τ	θ_1	θ_2
12	BB1	0.8	5.69	1.3
23	BB6	0.5	1.98	1.3
13|2	BB7	0.3	1.42	0.4

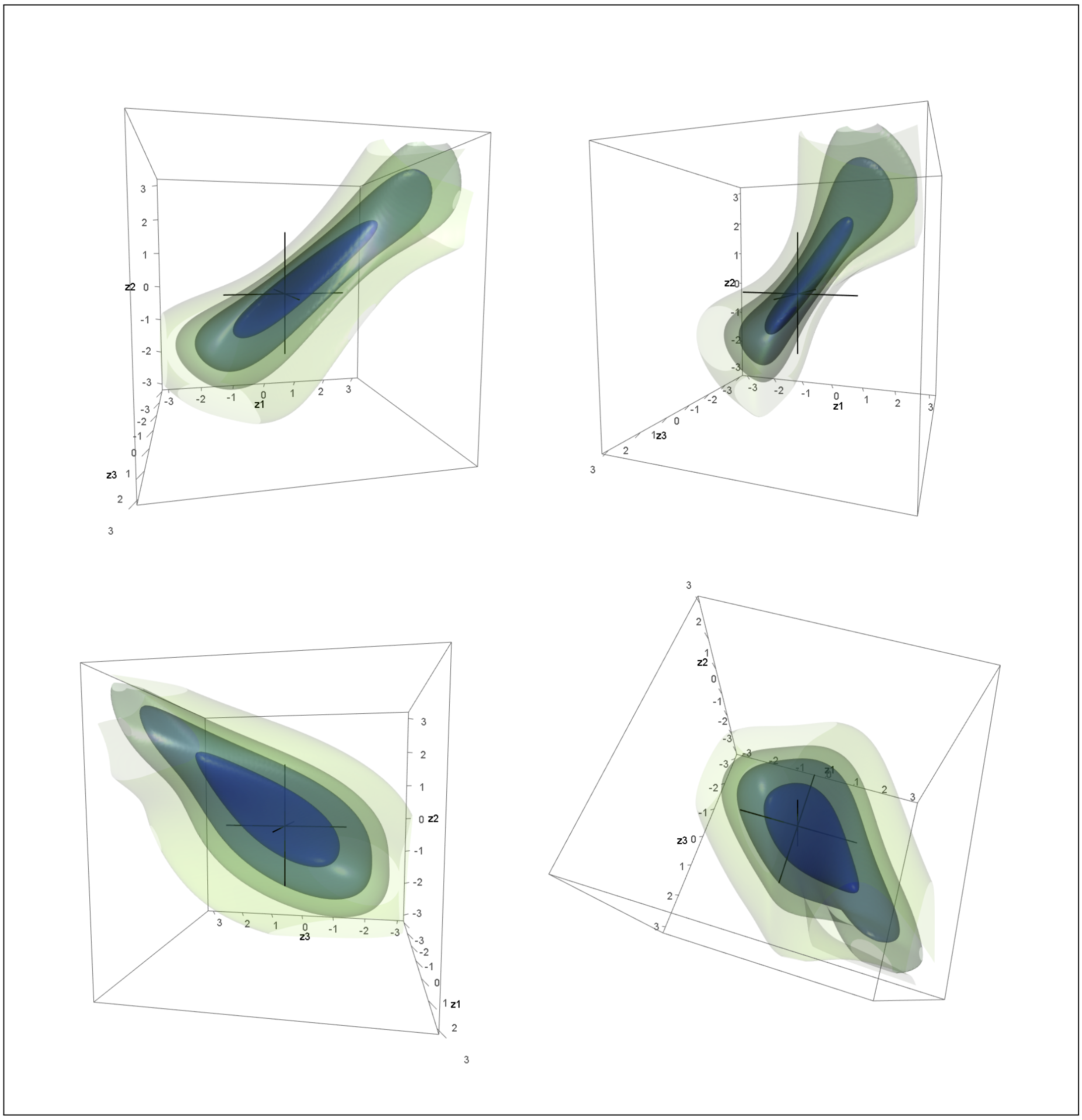

Figure 4.23:

Pair Copula	Family	τ	θ_1	θ_2
12	BB8	0.8	16.91	0.7
23	Tawn	0.5	2.16	0.9
13|2	Tawn2	0.3	1.47	0.9

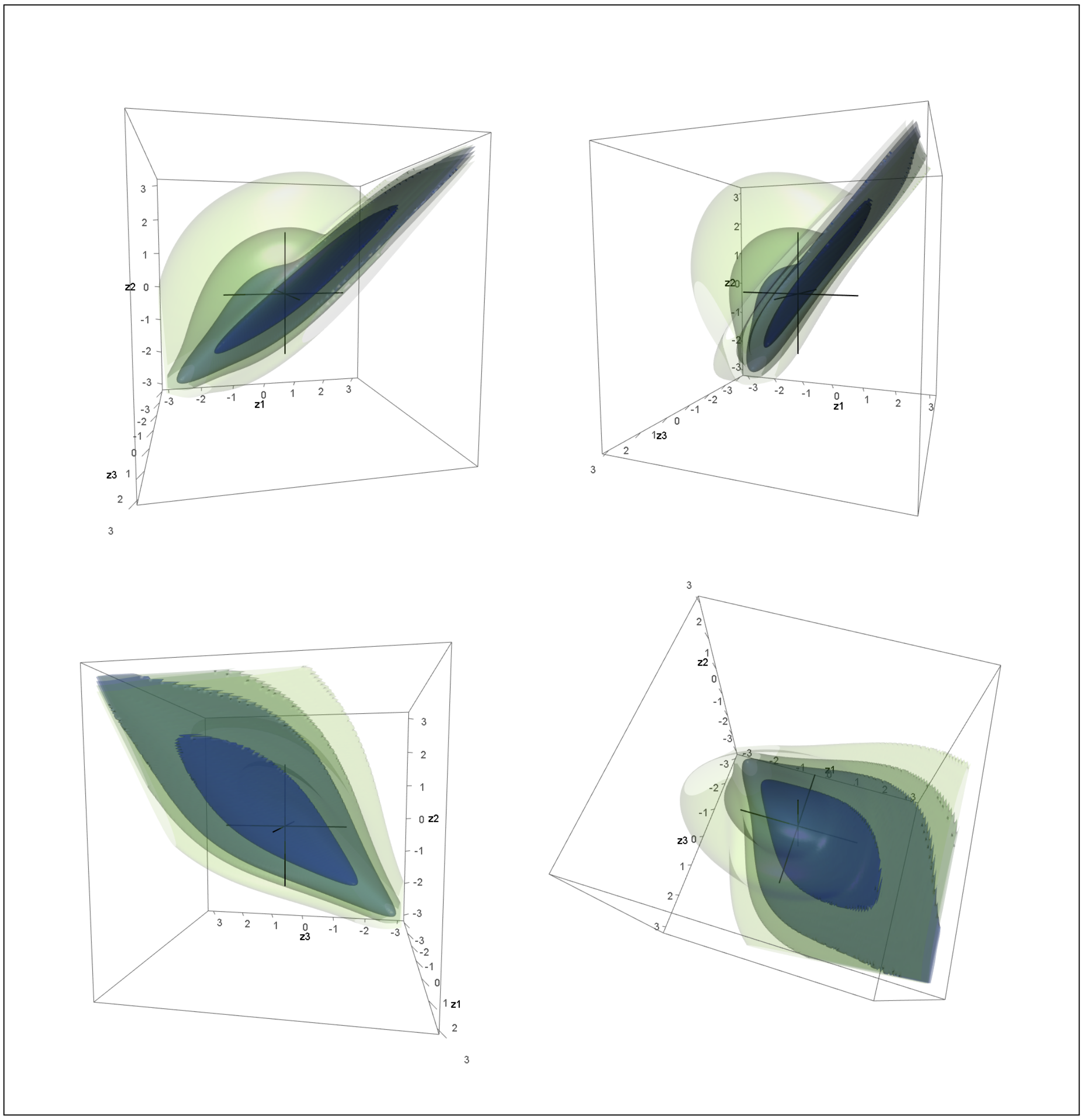

Figure 4.24:

Pair Copula	Family	τ	θ_1	θ_2
12	Tawn	0.8	8.28	0.9
23	BB1	0.5	1.08	1.3
13|2	BB8	0.3	3.2	0.7

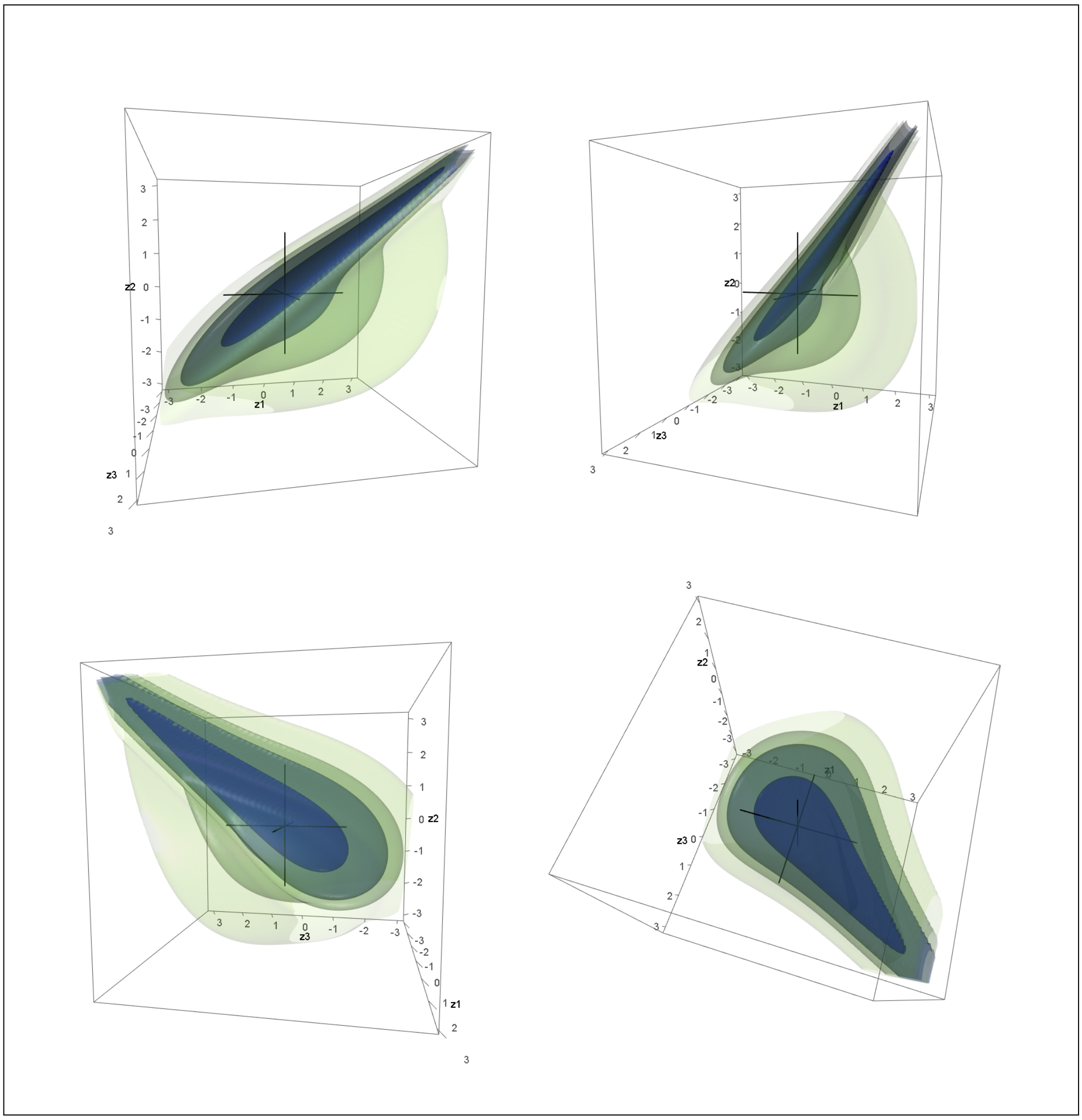

Figure 4.25:

Pair Copula	Family	τ	θ_1	θ_2
12	Tawn2	0.8	8.28	0.9
23	BB6	0.5	1.98	1.3
13|2	BB7	0.3	1.42	0.4

4.4.3 Scenarios with no Restriction of Copula Families

After having introduced each bivariate building block on its own and having built trivariate copulae with restrictions on the underlying copula families (due to the presentation in an ordered manner), we now are free to build our scenarios out of a big pool of bivariate copulae. But we stick to our standard for τ values.

We start with Figure 4.26 . This shows a trivariate copula constructed out of Frank, BB7 and Clayton copulae. Characteristic centralization of probability mass, modelled by a Frank copula, is visible in the upper pictures and upper tail dependence among second and third variable (BB7) can be seen in the lower left picture.

We continue with Figure 4.27 . Here the underlying building blocks consist out of a BB1, Student t and Joe copula. The picture on the upper right hand side visualizes the lower tail dependence of a BB1 copula best and the torn apart contour lines of a Student t copula can be observed in the lower pictures.

Now we want to a show a scenario in which a Gaussian copula occurs once again. It is used for modelling dependence among second and third variable in Figure 4.28 . The other copulae for this scenario are a Clayton and a Gumbel copula. The high lower tail dependence of a Clayton copula can be seen in the upper pictures.

Another interesting trivariate copula is visualized in Figure 4.29 . One can infer from the upper pictures that a Tawn copula type 1 is used for dependence among first and second variable. The lower left picture shows that a Student t copula describes the copula for the second building block and we see from the description that also a Gumbel copula is used for this trivariate copula.

Finally Figure 4.30 shows a trivariate copula out of a Frank, Clayton and BB6 copula as building blocks. The upper pictures visualize the concentration of probability mass in the centre (tightened contour lines at point of origin), which is modelled by the Frank copula. The Clayton copula causes the high lower tail dependence of second and third variable, which can be seen in the pictures on the lower side of the figure.

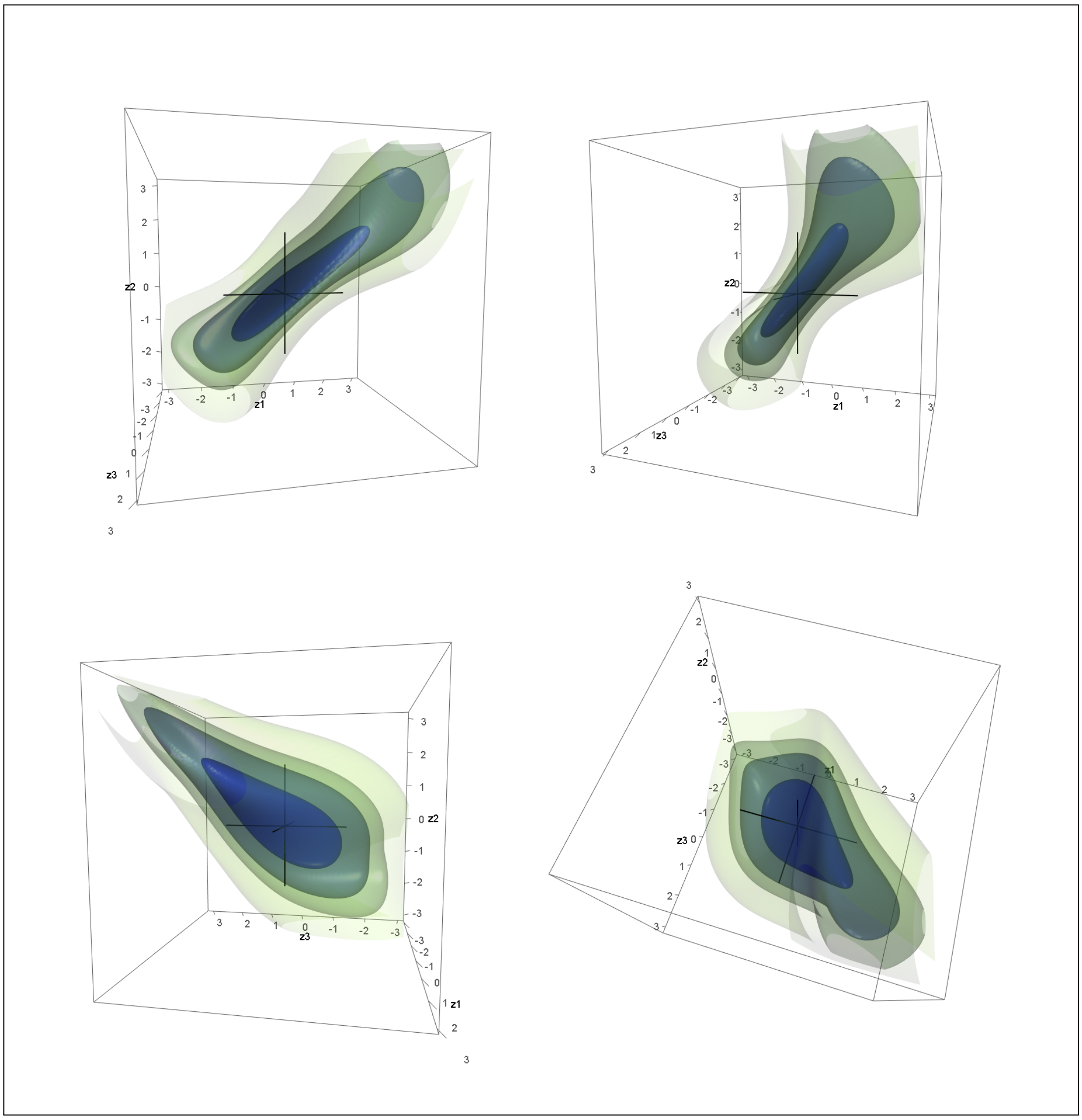

Figure 4.26:

Pair Copula	Family	τ	θ_1	θ_2
12	F	0.8	18.19	0
23	BB7	0.5	2.5	0.4
13|2	C	0.3	0.86	0

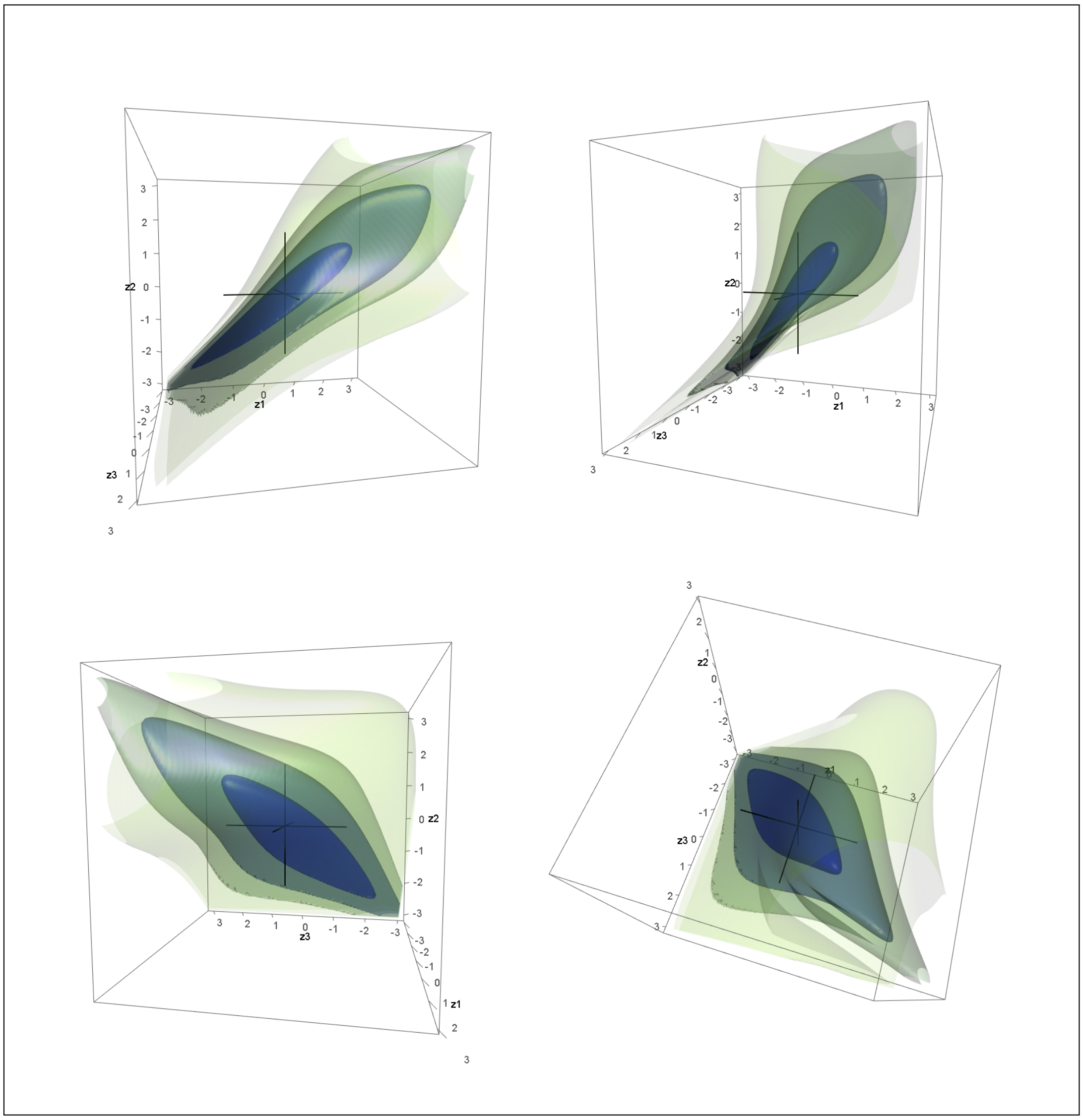

Figure 4.27:

Pair Copula	Family	τ	θ_1	θ_2
12	BB1	0.8	5.69	1.3
23	t	0.5	0.71	3
13|2	J	0.3	1.77	0

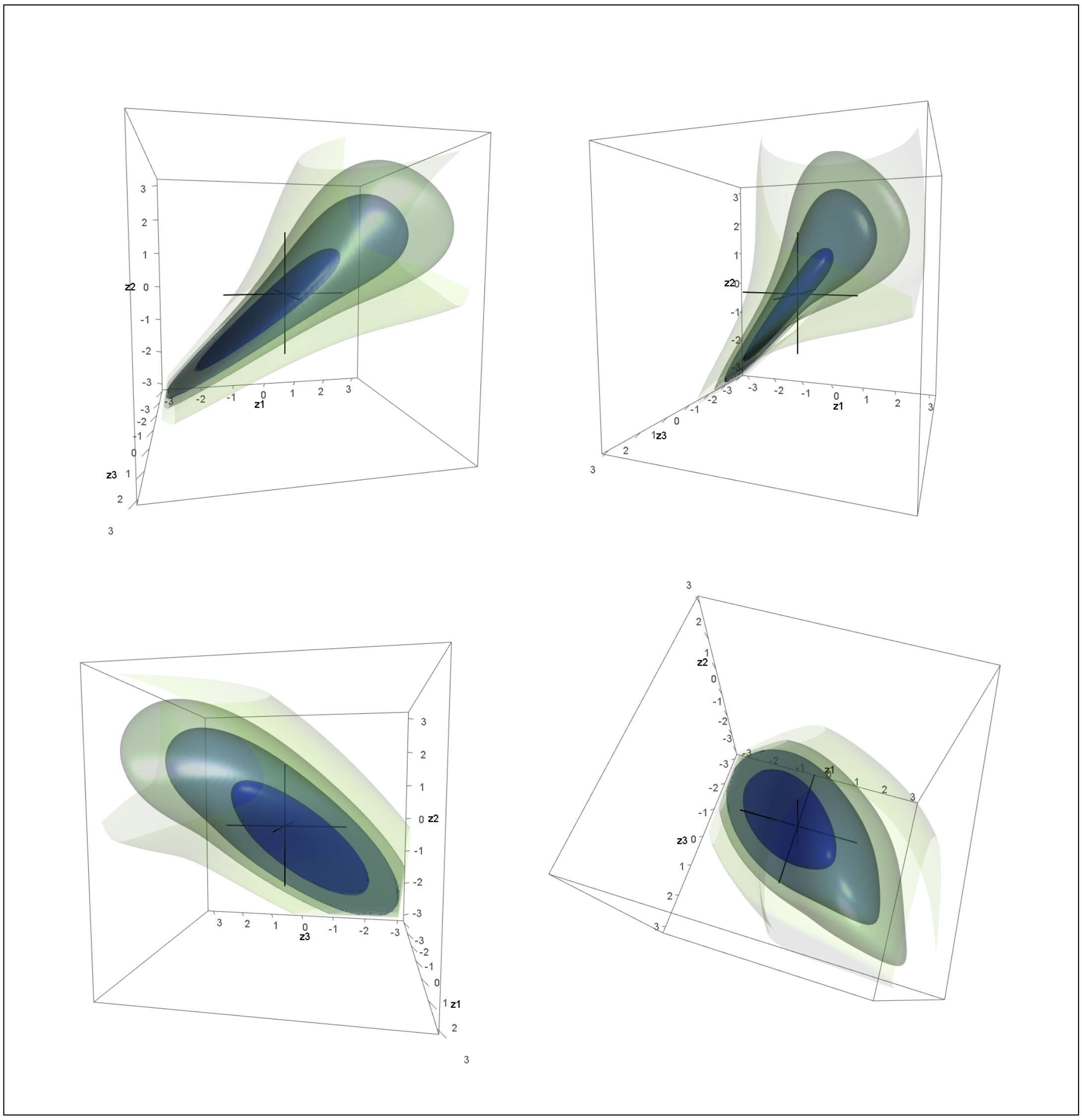

Figure 4.28:

Pair Copula	Family	τ	θ_1	θ_2
12	C	0.8	8	0
23	N	0.5	0.71	0
13|2	G	0.3	1.43	0

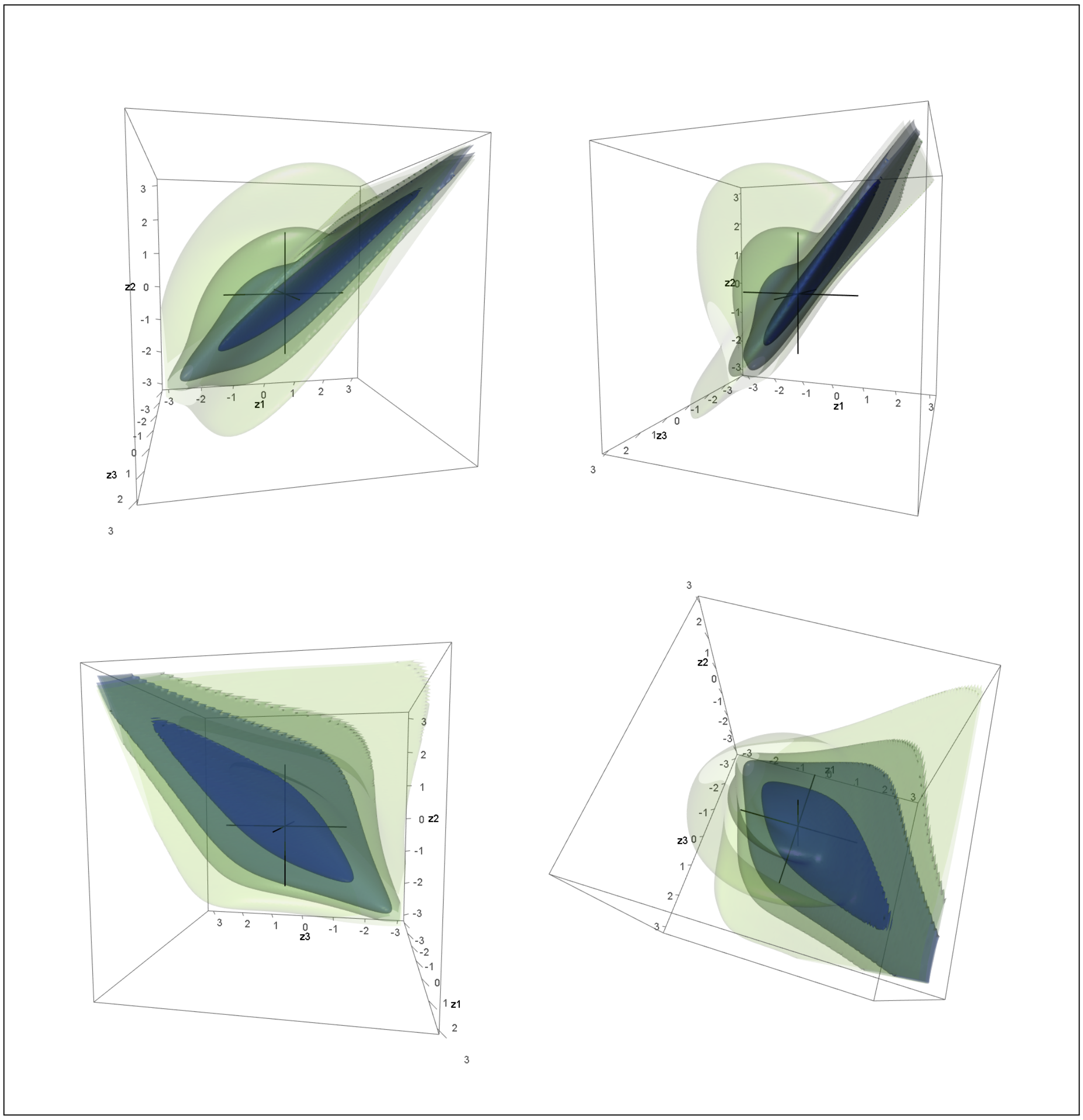

Figure 4.29:

Pair Copula	Family	τ	θ_1	θ_2
12	Tawn	0.8	8.28	0.9
23	t	0.5	0.71	3
13|2	G	0.3	1.43	0

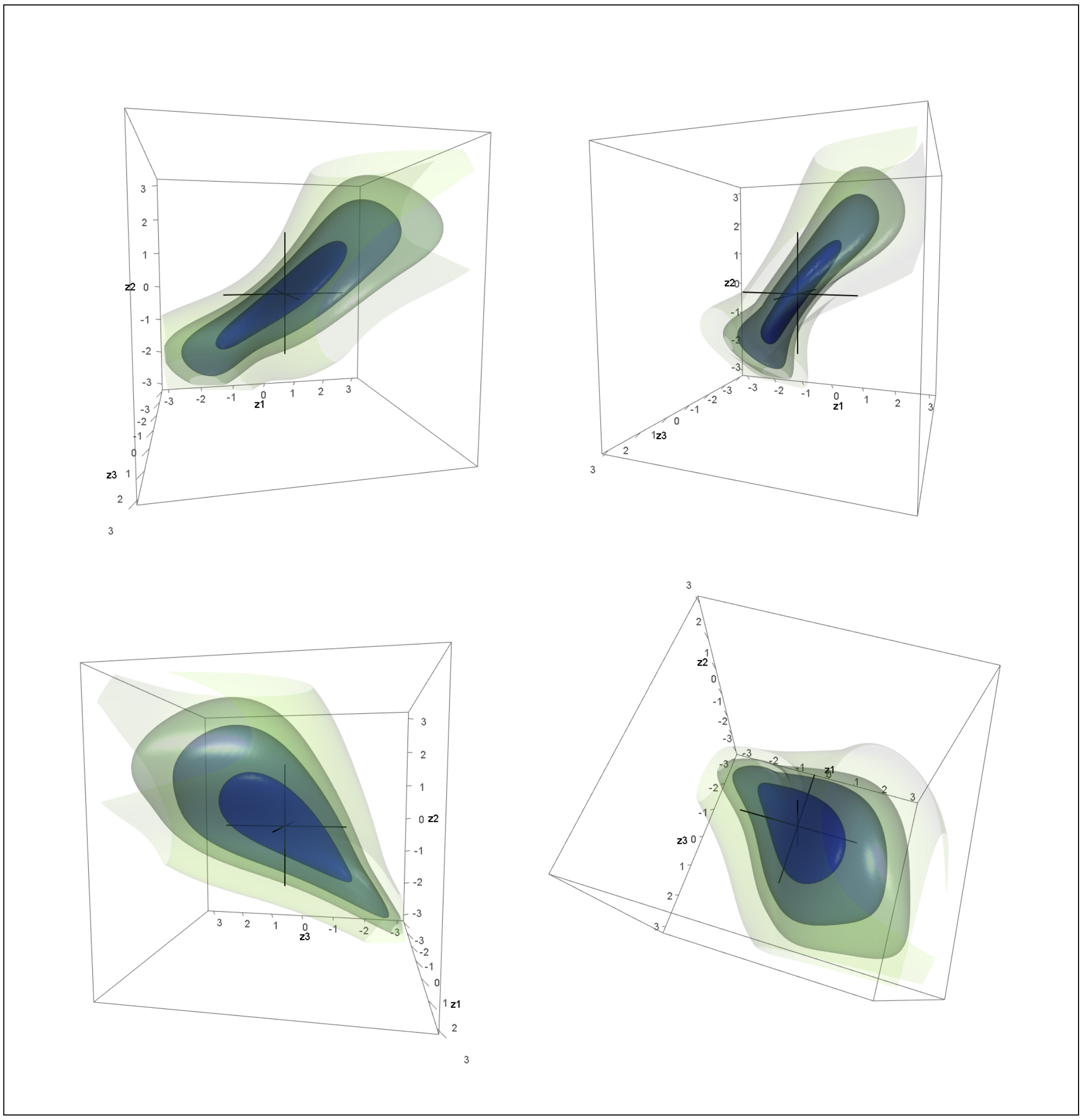

Figure 4.30:

Pair Copula	Family	τ	θ_1	θ_2
12	F	0.8	18.19	0
23	C	0.5	2	0
13|2	BB6	0.3	1.17	1.3

4.4.4 Scenarios with no Restriction on τ Values

Now we leave our standard τ setting of 0.8 , 0.5 and 0.3 for the building blocks and show the great variety of dependence structures that can be modelled using copulae.

Figure 4.31 is interesting because of the twisted shape, best visible in the lower right picture. This is caused by high dependence of the third building block modelled via a Frank copula. The Clayton and Gumbel copula used for the other building blocks model low dependence and thus the contour lines are pulled apart.

The pictures in Figure 4.32 seem to do not belong together. The ones on the upper side suggest a rectangular shape, which is rebutted by the pictures on the lower side. The picture on the lower right hand side may actually prefer a somehow triangular shape. This shows the interesting interplay of a Student t, Clayton and Gaussian copula as building blocks. The three dimensional shape of the scenario can be understood best by rotating it on ones own after having visualized the scenario via *R*. It can be found under the label “vis+NRT2” in the scenario data set. Another interesting aspect is that all contour lines, even the one of the 0.75 quantile, can be drawn in closed shape. As dependence is low among the variables, the probability mass is highly centralized. One can also compare this to the independence copulae shown in Figure 4.7 .

In Figure 4.33 the picture on the lower left hand side again seems not fitting to the other pictures. As this is the snapshot taken from the side and it visualizes the influence of Gumbel copula, no rectangular shape can be observed there. This property of a Student t copula, used for both remaining building blocks, can be seen in all the other pictures.

Figure 4.34 shows a trivariate copula constructed out of a Gaussian, Clayton and BB1 copula. One realizes that the contour line of the 0.75 quantile is truncated and so probability mass is decentralized. The lower right hand picture shows high dependence among first and third variable as the contour lines are tightened together.

Finally Figure 4.35 visualizes a trivariate copula built out of a Gumbel, Student t and BB7 copula. The contour lines are pulled apart due to general rather low dependence. The characteristic rectangular shape of a Student copula can be observed best in the lower left picture and a Gumbel copula, with its typical upper tail dependence, can be divined in the upper left picture.

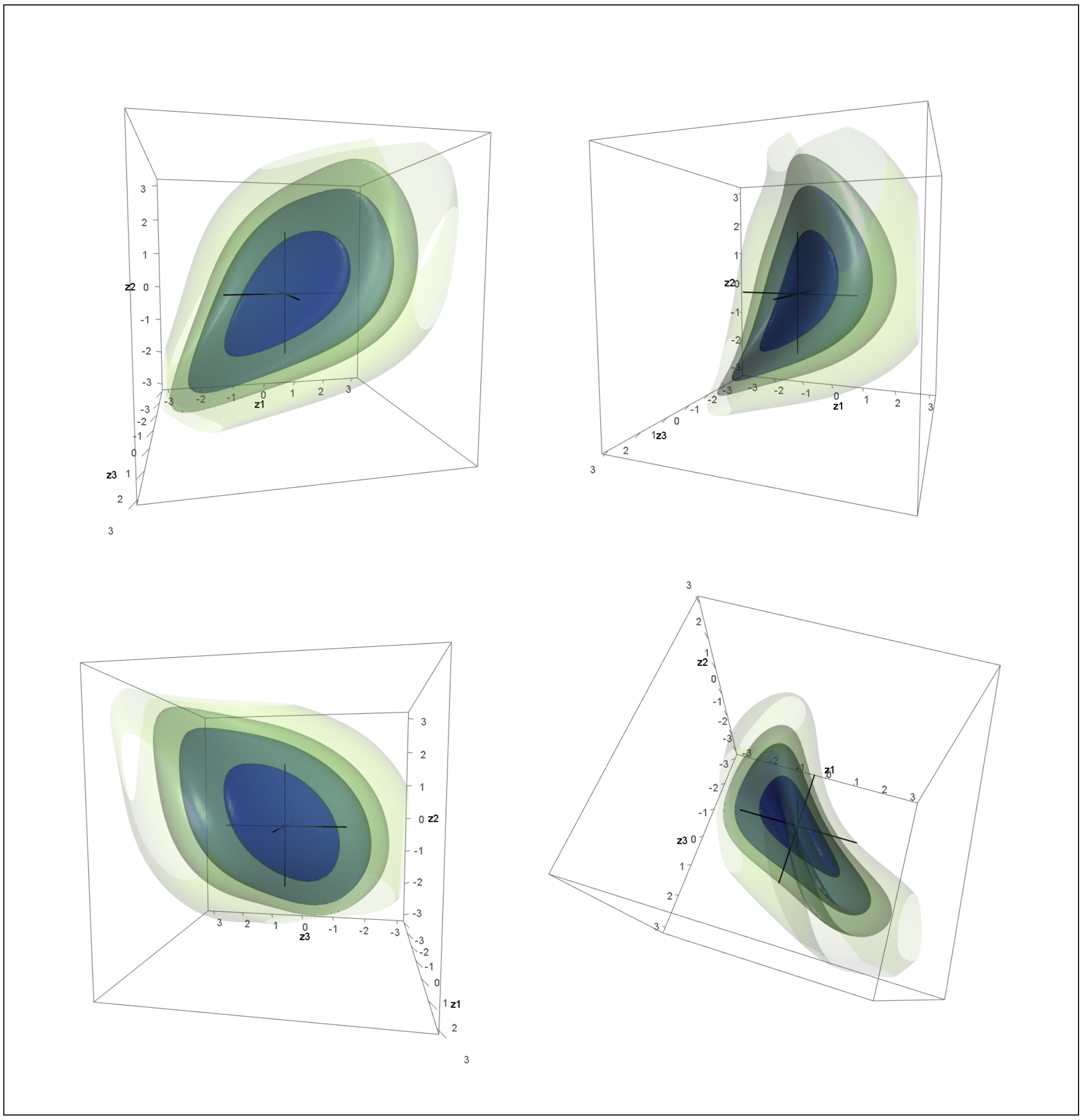

Figure 4.31:

Pair Copula	Family	τ	θ_1	θ_2
12	C	0.3	0.86	0
23	G	0.2	1.25	0
13|2	F	0.8	18.19	0

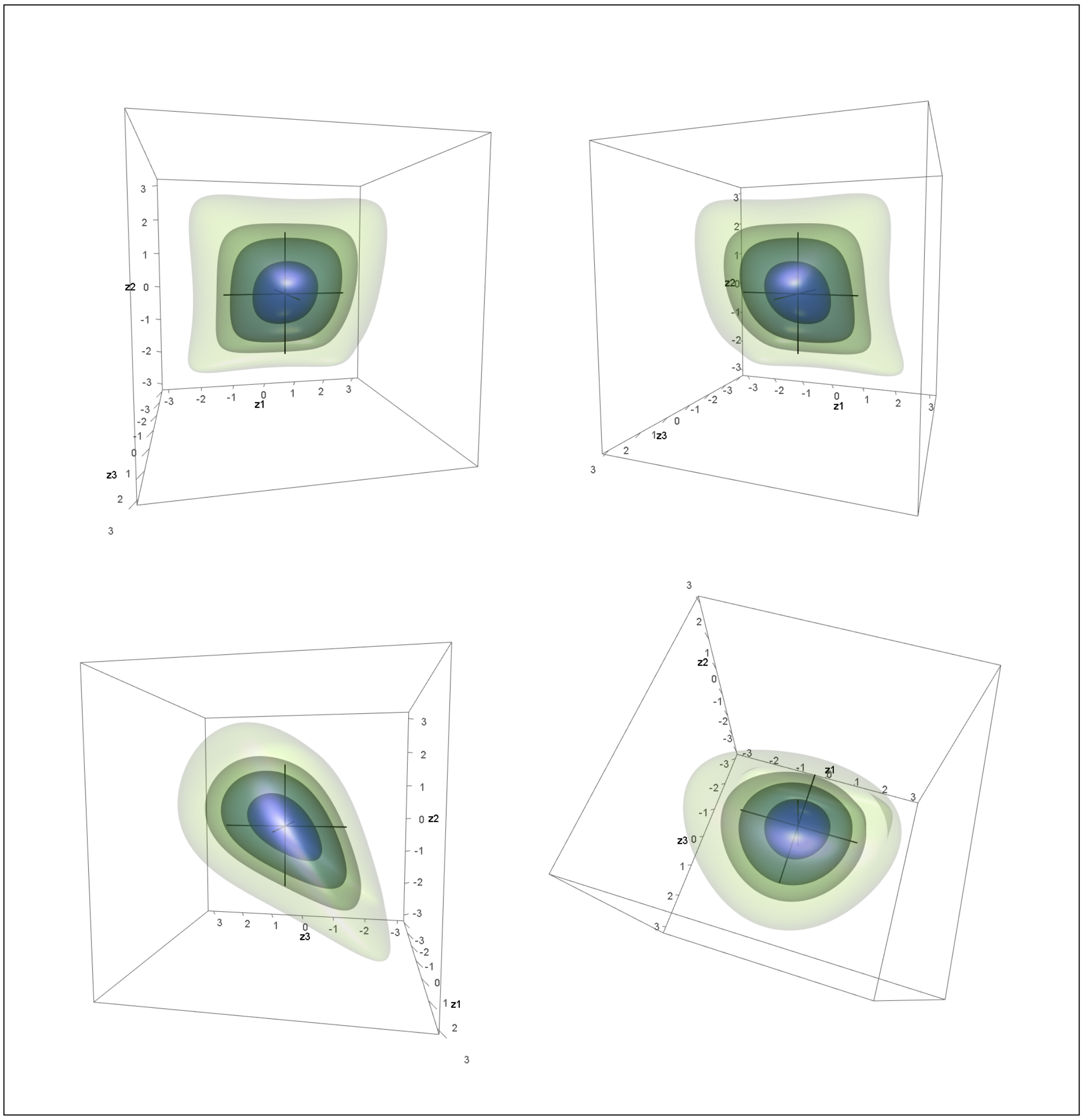

Figure 4.32:

Pair Copula	Family	τ	θ_1	θ_2
12	t	0	0	3
23	C	0.33	0.99	0
13|2	N	0	0	0

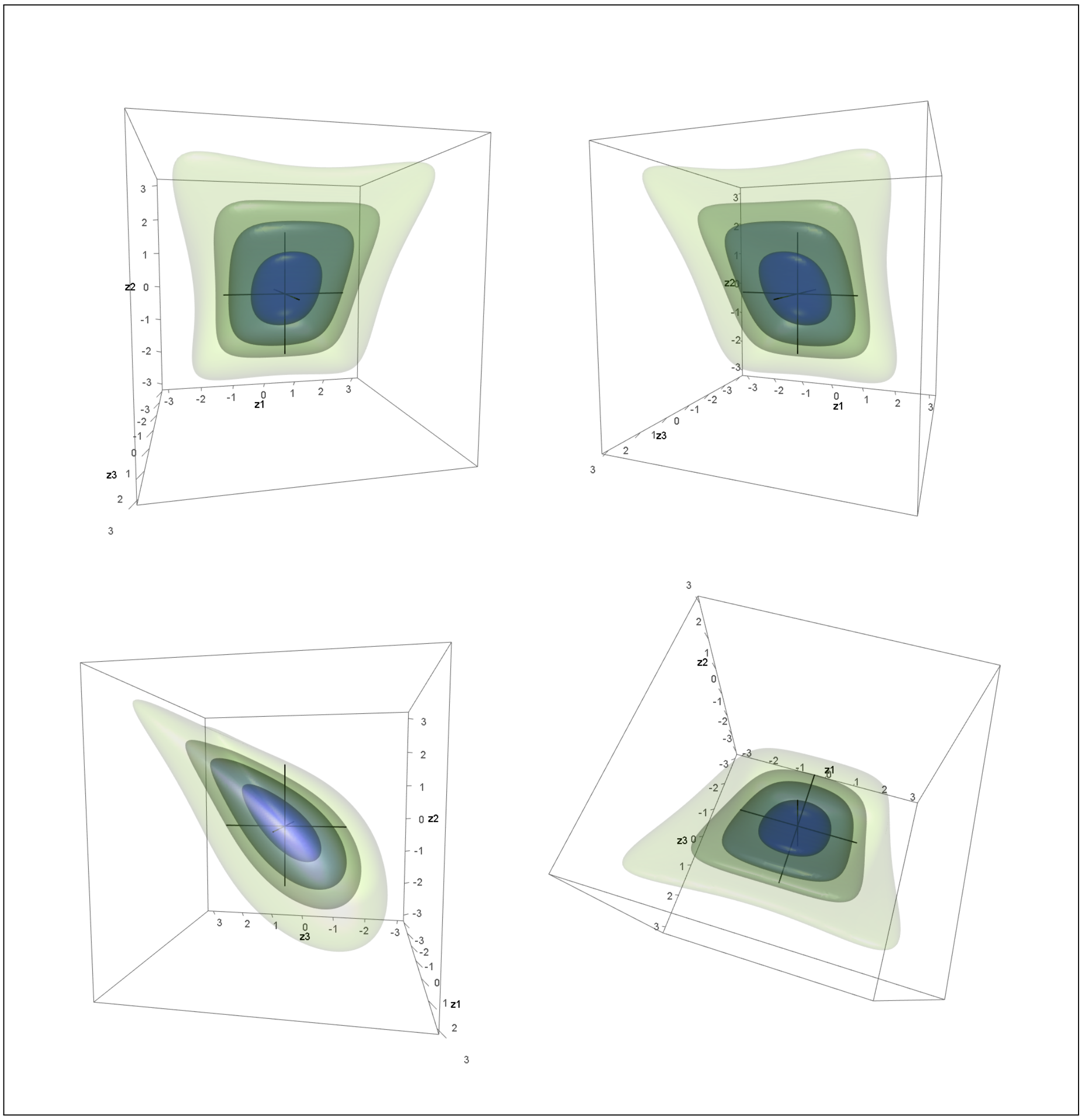

Figure 4.33:

Pair Copula	Family	τ	θ_1	θ_2
12	t	0	0	3
23	G	0.5	2	0
13|2	t	0	0	5

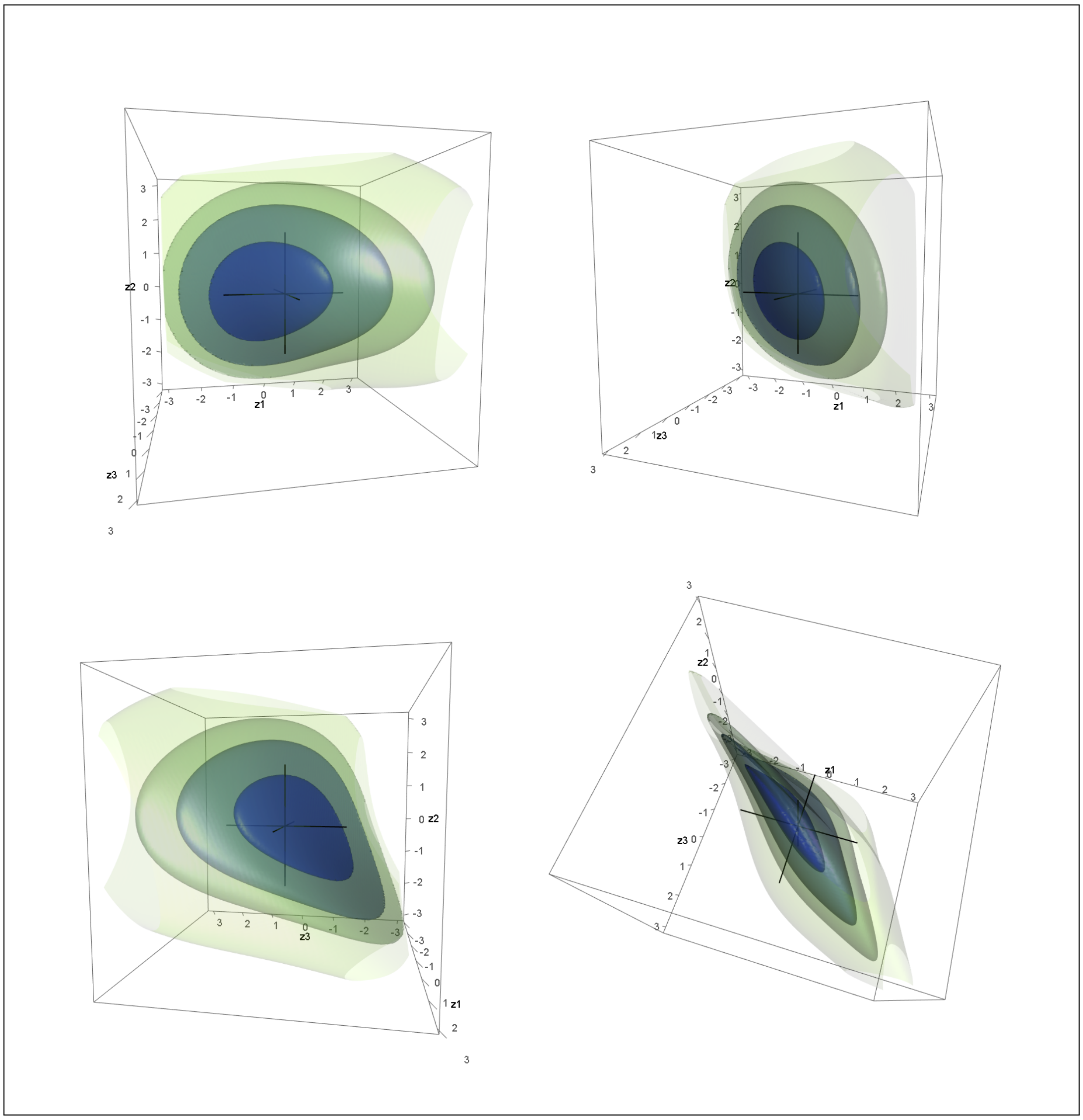

Figure 4.34:

Pair Copula	Family	τ	θ_1	θ_2
12	N	0	0	0
23	C	0.33	0.99	0
13|2	BB1	0.75	2	2

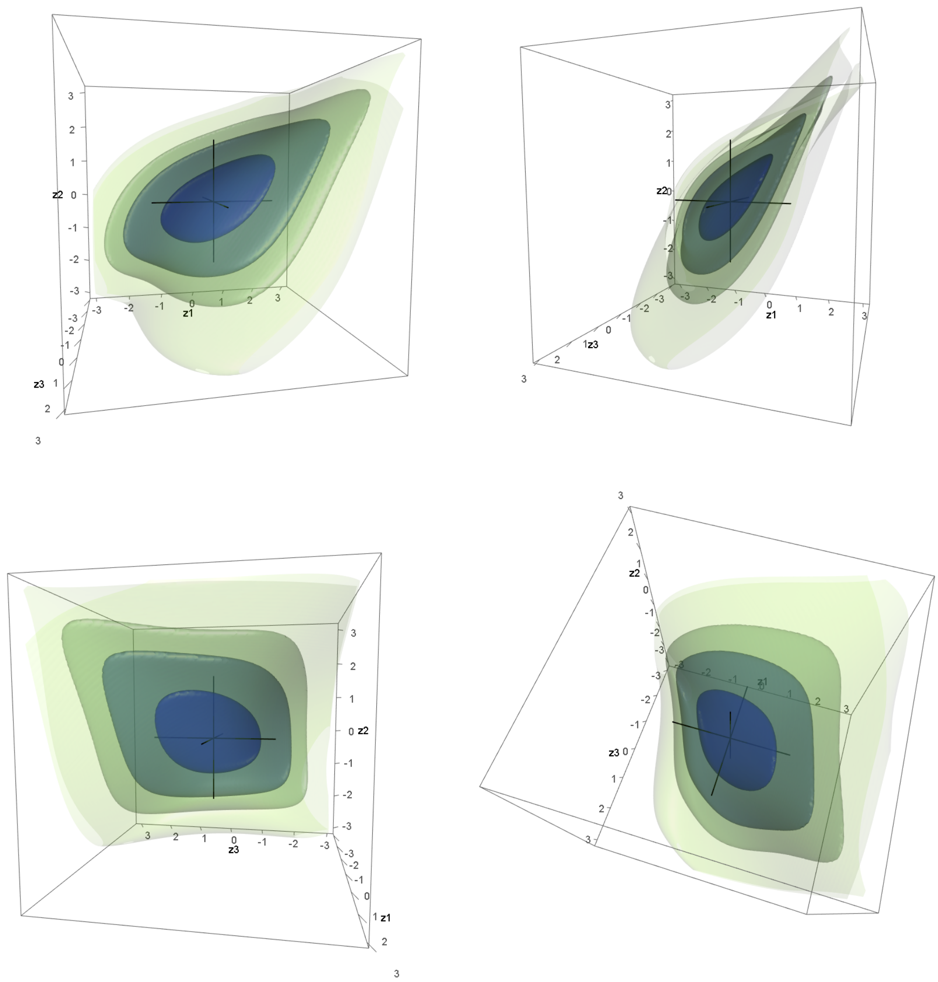

Figure 4.35:

Pair Copula	Family	τ	θ_1	θ_2
12	G	0.5	2	0
23	t	0	0	3
13|2	BB7	0.6	2.22	2

4.4.5 Scenarios in Mai and Scherer (2012)

Next we want to visualize scenarios given in Table 5.2 in Mai and Scherer (2012) on page 222. These scenarios have really interesting shapes and we need lots of points to visualize them properly. But even with many evaluations of the probability density the third scenario still caused heavy numerical problems and thus we skip showing it in the thesis. Nevertheless it is available in the scenario data set with the label “vis+Mai3”.

In Figure 4.36 the underlying copula is built out of a Gumbel and two Clayton copulae. The Clayton copula for the bivariate building block of second and third variable is rotated by 90 degrees. This is denoted by C90 and details on rotation can be found in Section 3.5 of Czado (2013) on page 30 et seq. Strong dependence among first and second variable visualizes the typical shape of a Gumbel copula as in Figure 2.4 and can be seen in the upper left hand picture. Another interesting feature of this scenario is the negative dependence of second and third variable.

Figure 4.37 shows a scenario with two Student t copulae and one Gumbel copula as building blocks. The resulting contour levels are very flat and torn apart as dependence among second and third variable is the lowest.

Very high dependence in all building blocks characterizes the scenario in Figure 4.38 . Two Frank copulae and one Clayton copula were used to model the situation. High dependence is visualized in tight contours and even the 0.99 quantile can not be drawn in closed shape as one recognizes in the lower right hand picture. So lots of probability mass is contained in the tails. The interesting S-shape can be seen best in the upper right hand picture. In this figure we also recognize this dependence structure to cause numerical problems, as the “fingers” towards the lower panel show.

All in all these three scenarios have very different shapes and thus demonstrate a great flexibility when modelling dependency with copulae.

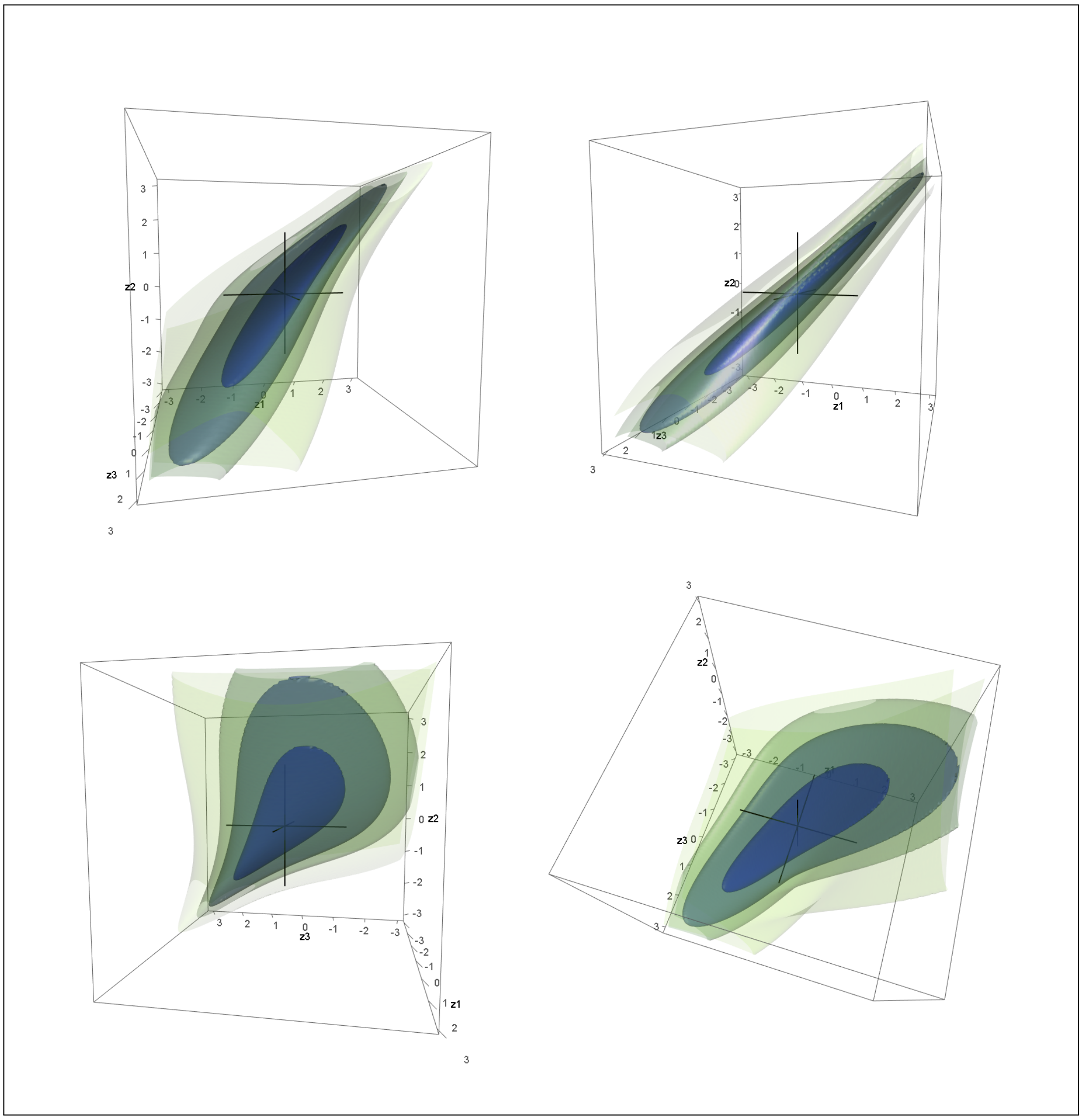

Figure 4.36:

Pair Copula	Family	τ	θ_1	θ_2
12	G	0.8	5	0
23	C90	-0.54	-2.35	0
13|2	C	0.26	0.7	0

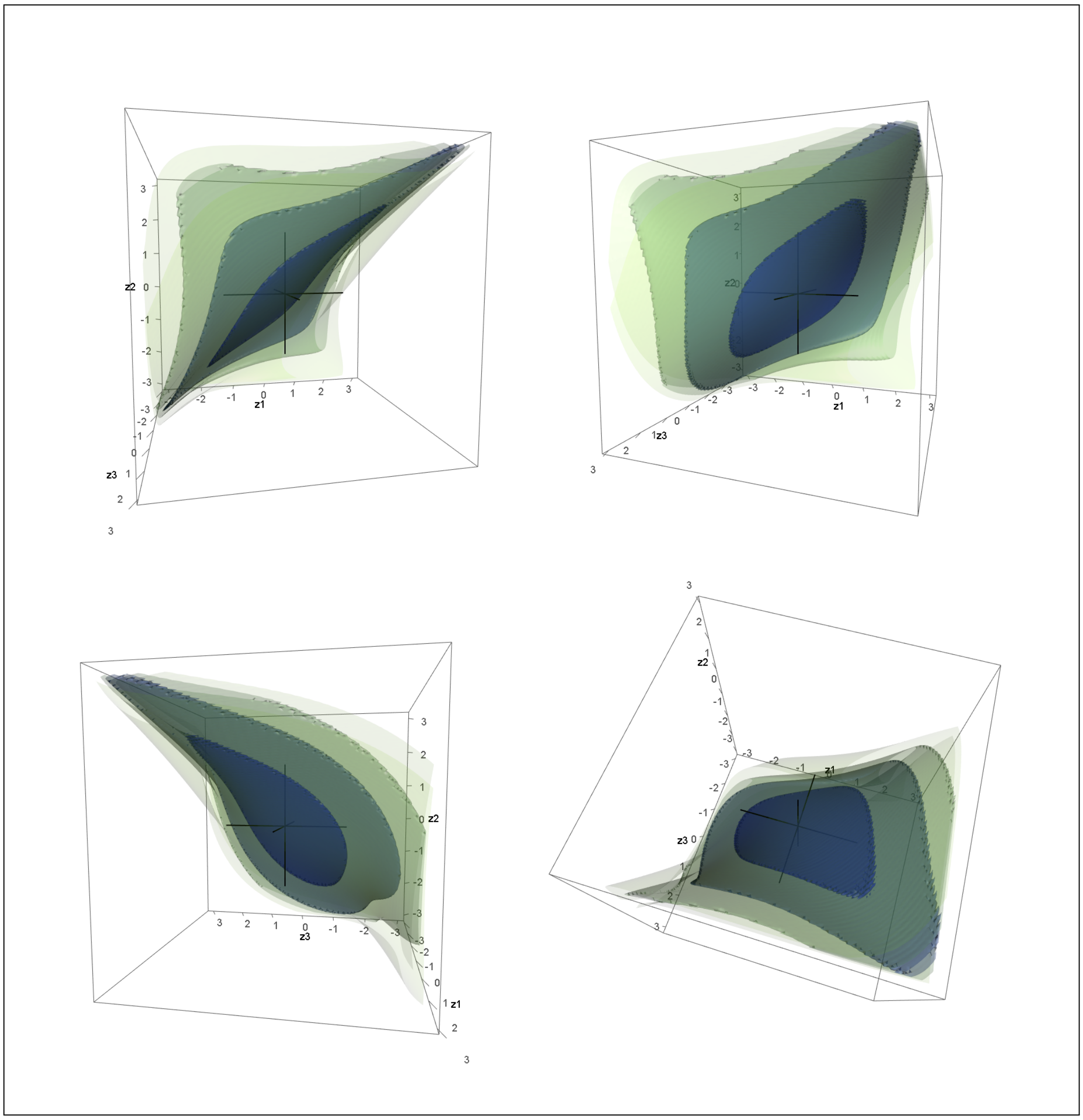

Figure 4.37:

Pair Copula	Family	τ	θ_1	θ_2
12	t	0.59	0.8	2.1
23	G	0.43	1.75	0
13|2	t	-0.8	-0.95	2.5

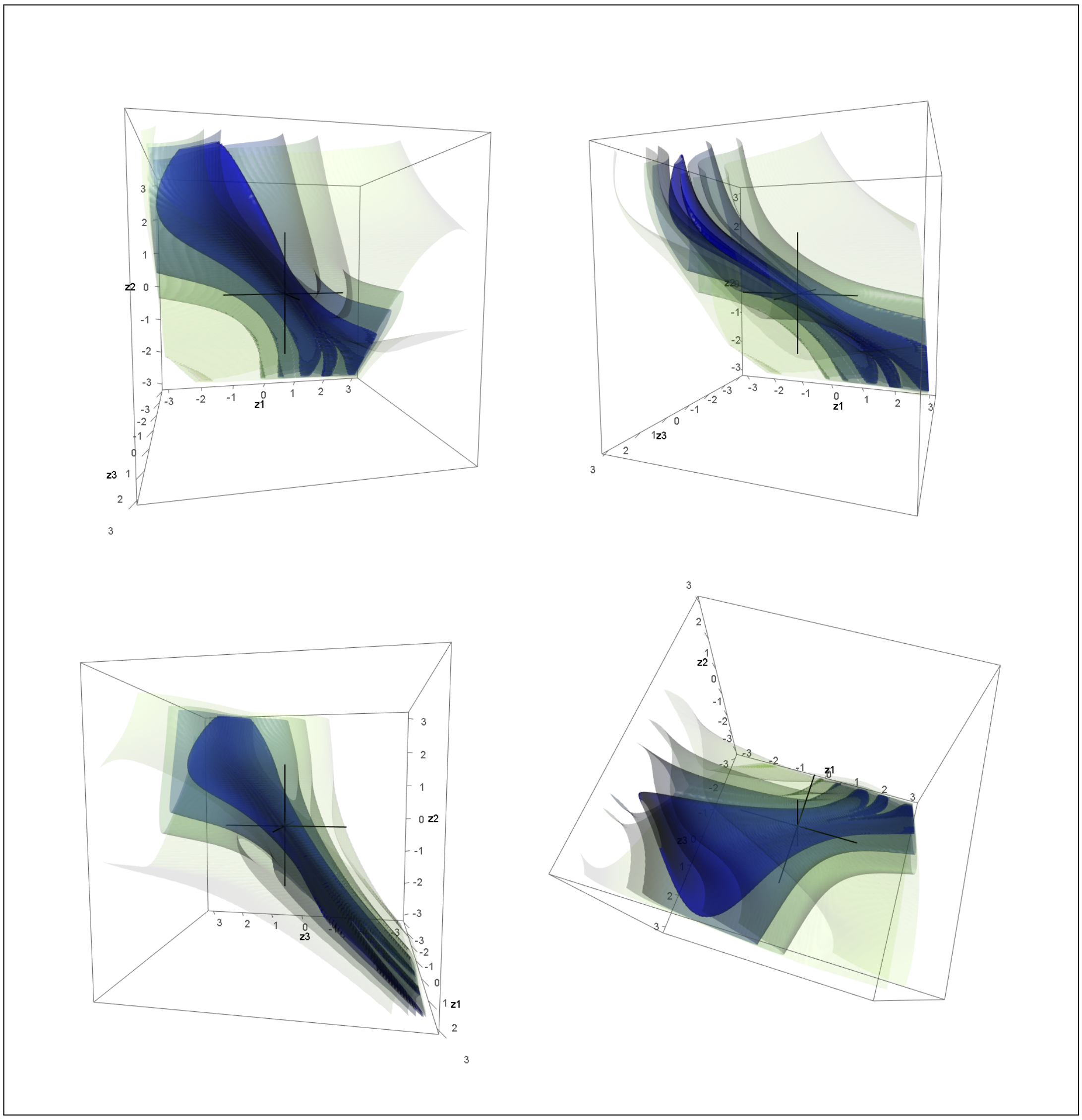

Figure 4.38:

Pair Copula	Family	τ	θ_1	θ_2
12	F	-0.89	-34.64	0
23	C	0.91	20.22	0
13|2	F	0.89	34.64	0

4.5 Visualization of Uranium Exploration Dataset

Finally we also want to apply our visualization method to a given dataset. We will examine the `Uranium` dataset given in the **copula** package and want to study dependence among the variables “Co, Sc and Ti” as suggested by my supervisor Prof. Claudia Czado. For this we roughly follow the process given in Brechmann and Schepsmeier (2013) but with the up-to-date methods of the *R* package **VineCopula**. In this section we directly give the corresponding code to reveal the procedure.

4.5.1 Getting Uniform Data

We first load the relevant dataset and transform it to uniform data. To do so we use the pseudo-observations function *pobs* from **VineCopula** and save the output as a data frame. This uniform data will allow us to do Copula calculations later on.

```
data("uranium", package = "copula")
uranium <- uranium[names(uranium) %in% c("Co", "Sc", "Ti")]

uranium <- data.frame(pobs(data.frame(Co = uranium$Co, Sc =
   uranium$Sc, Ti = uranium$Ti)))
```

4.5.2 First Impression of Data

Next we have to select the structure for the copula. For a first impression of the data we draw a pairs plot with scatter plots above and contour plots with standard normal margins below the diagonal.

```
wd <- getwd()
sd <- file.path(wd, "..", "R-Pictures", "Scenarios",
   "vis+Uranium")
if (!file.exists(sd)) dir.create(sd)
setwd(sd)

pdf(file = "pairs-copula_uranium.pdf")
pairs.copuladata(uranium, gap=0)
dev.off()

setwd(wd)
```

On the diagonal panel of Figure 4.39 one can see histograms of our transformed data set, which are now predominant uniformly distributed. So our transformation above worked fine and copula calculations can be done. The contour plots on the lower panel show that there is no great tail dependence, so we would not model this via a Clayton or Gumbel copula for example. This can also be seen on the scatter plots on the upper panel. The τ values on the upper panel show the lowest dependence is between Co and Ti. Maximal dependence is with 0.54 on a medium level.

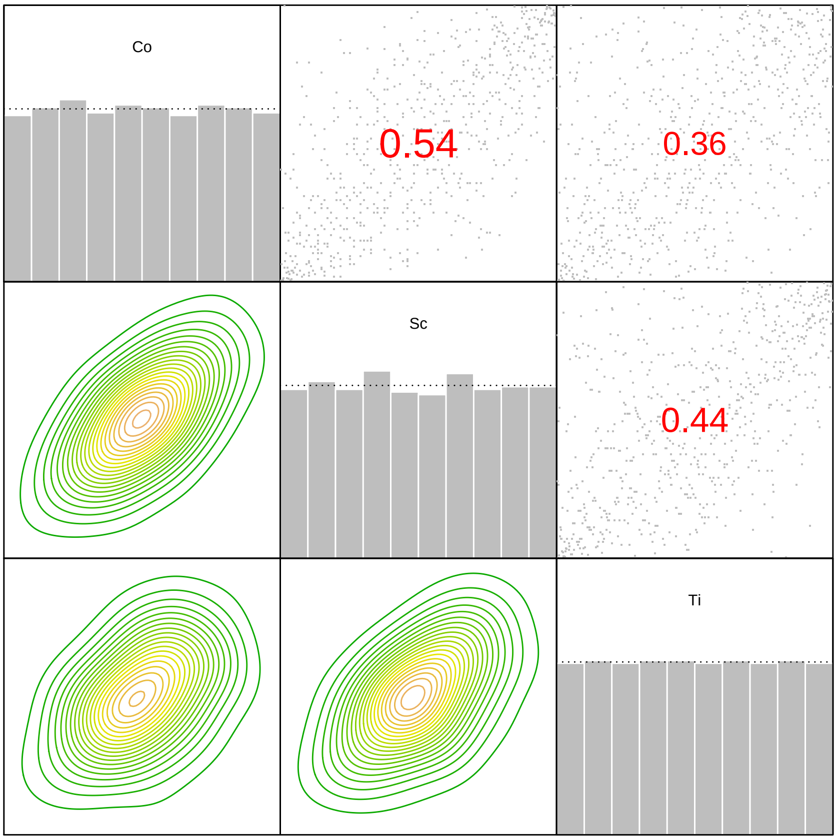

Figure 4.39: First impression of Uranium data set after getting uniform data. Histograms on the diagonal, contour plots below the diagonal and scatter plots above. Axes are from -3 to 3.

4.5.3 Fit Copula Model

Now we want to fit a copula model to our data set, meaning to select structure, copula families and corresponding parameters that suits the data best. The function *RvineStructureSelect* does the job.

```
( uraniumRVM <- RVineStructureSelect(uranium) )
```

```
R-vine matrix:
```

```
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    3    2    0
[3,]    2    3    3
```

```
Where
1 <-> Co
2 <-> Sc
3 <-> Ti
```

So we have the structure 13|2, 12 and 23 (which is our standard structure for all scenarios), each modelled by t copulae. Other quantities as family, first and second parameter are also contained in the list "uraniumRVM" and are presented in the caption of Figure 4.40.

4.5.4 Visualizing the Model

It is left to visualize this three dimensional model using our visualization functions. So in Figure 4.40 one sees the typical diamond shape of t copulae. The dependence among first and third variable is quite low, as can be seen by the corresponding τ value, but also in the lower right hand picture. The contour lines in this picture are not as tightened together as they are in the other pictures and also the contour line of the 0.99 quantile is closer to the centre.

```
wd <- getwd()
sd <- file.path(wd, "..", "R-Pictures", "Scenarios")
if (!file.exists(sd)) dir.create(sd)

folder <- file.path(sd, "vis+Uranium")
if (!file.exists(folder)) dir.create(folder)
setwd(folder)

z1 <- seq(-3,3,by=0.2)
z3 <- z2 <- z1

f_uranium = f_Z(z1, z2, z3, uraniumRVM)

vis3Dvine(f_uranium)


fam12 <- round(uraniumRVM$family[3,1], 2)
fam23 <- round(uraniumRVM$family[3,2], 2)
fam13.2 <- round(uraniumRVM$family[2,1], 2)
par12 <- round(uraniumRVM$par[3,1], 2)
par23 <- round(uraniumRVM$par[3,2], 2)
par13.2 <- round(uraniumRVM$par[2,1], 2)
```

```
par2_12 <- round(uraniumRVM$par2[3,1], 2)
par2_23 <- round(uraniumRVM$par2[3,2], 2)
par2_13.2 <- round(uraniumRVM$par2[2,1], 2)
tau12 <- round(BiCopPar2Tau(fam12, par12, par2_12), 2)
tau23 <- round(BiCopPar2Tau(fam23, par23, par2_23), 2)
tau13.2 <- round(BiCopPar2Tau(fam13.2, par13.2, par2_13.2), 2)
label <- "vis+Uranium"

scenarioToLatex(fam12, fam23, fam13.2,
                par12, par23, par13.2,
                par2_12, par2_23, par2_13.2,
                tau12, tau23, tau13.2,
                label)

# put label in a tex file
write(paste0("\\label{", label, "}"), file = "00label.tex")

# put one text file with the name in the folder
write("", file = paste0("Uranium", ".txt"))

write(
  paste(
    paste0("\\pageref{", label, "}"),
    BiCopName(fam12), tau12, par12, par2_12,
    BiCopName(fam23), tau23, par23, par2_23,
    BiCopName(fam13.2), tau13.2, par13.2, paste0(par2_13.2,
       "\\\\")
    , sep = " & "
  ), file = file.path(sd, "ScenarioTable.tex"), append = TRUE )
write("\\end{longtable}  }",
      file = file.path(sd, "ScenarioTable.tex"), append = TRUE )
write("The \\enquote{NA} between page 84 and 85 corresponds to
   the third scenario of \\cite{Mai.2012} which causes strong
   numerical problems and thus is not shown in the thesis.",
      file = file.path(sd, "ScenarioTable.tex"), append = TRUE )
setwd(wd)
```

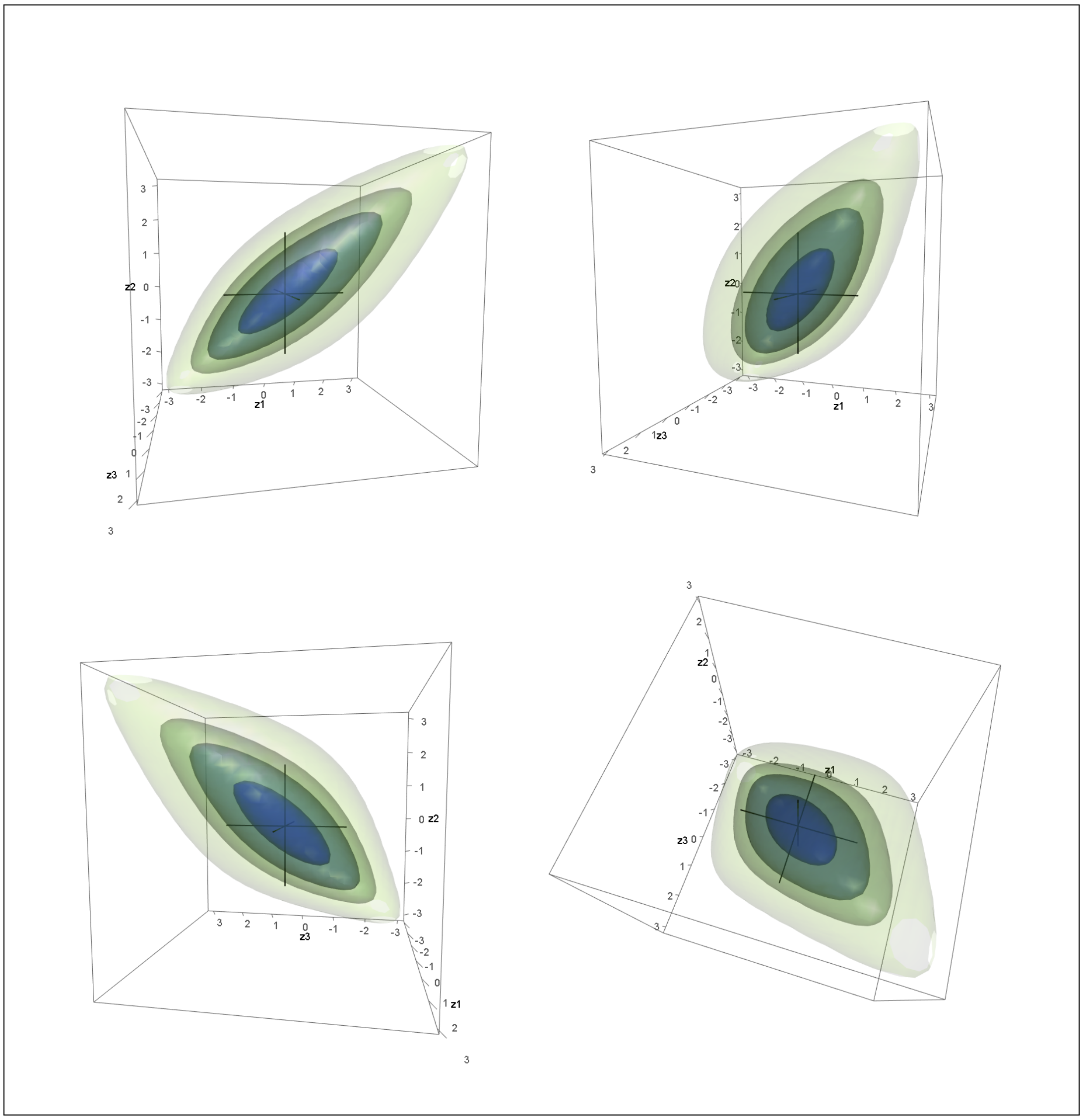

Figure 4.40:

Pair Copula	Family	τ	θ_1	θ_2
12	t	0.53	0.74	8.02
23	t	0.43	0.62	5.93
13|2	t	0.08	0.13	5.65

Chapter 5

Conclusion

The purpose of this thesis was the visualization of trivariate copulae. To achieve this, we used some copula theory and constructed trivariate copulae out of bivariate building blocks, remember the pair copula construction. We did the visualization by implementing this theory with the statistical software *R*.

In Chapter 1 we got used to the idea of visualizing dependence among three random variables using contour lines via the normal distribution. We resumed basic facts on it and made our way from the univariate case to the trivariate normal distribution.

Chapter 2 provided some theory of copulae and dependence measures. It also contained a section on bivariate copula classes, where elliptical, Archimedean and Tawn copulae were shown.

As our aim was to visualize trivariate copulae, we took a closer look on the trivariate Gaussian copula in Chapter 3 . This chapter also presented pair copula construction and its application to the trivariate case.

The main part of the thesis was Chapter 4 , which first showed the workflow of our implementation in *R*. Then we used these tools to visualize trivariate Gaussian and Student t copulae and lots of further scenarios in an ordered manner. We finally applied our functions in practice by examining dependence of the uranium data set. All the code is also given in the *R* package **copulaSG**.

So with this thesis we provided another tool to visualize data sets. In general, data gets more important all the time and we have the opportunities to store enormous amounts of information. To draw conclusions from this one may use statistical theory and visualization. Both can be done by statistical software like *R*, which is further developed by a great community. When calculating with copulae, the *R* package **VineCopula** is an excellent choice and could contain our tools to visualize trivariate copulae in the future.

Appendix A

R-Code Normal Distribution

A.0 General Settings

In this chapter we want to show the *R* code for plotting the probability density and contour lines of one and two dimensional standard normal distributions, which we showed in Chapter 1. Therefore we need the *R* package **mvtnorm**.

```
library(mvtnorm)
```

Before getting to the visualization, we first set up the folder where to save the plots and create it if it does not already exist. Later on, we will use the `pdf` device, which will by default create quadratic vector graphics that are independent of scaling.

A.1 The Univariate Normal Distribution

First we set up the contour levels by calculating the specified quantiles and provide the colours in which to plot them. We use a self defined colour palette from green to blue throughout the thesis. After specifying the parameters for the device to draw at, we can start plotting the probability density of a one dimensional standard normal distribution (default of *dnorm*). As we want to show the contour lines, we have to get the points at which the contour levels are reached. To do so, we specify the inverse function and save the corresponding points in the variable `contourPoint`. Now it is time to add the points to the graphic with the right colours.

```
sd <- file.path(getwd(), "..", "R-Pictures")
if (!file.exists(sd)) dir.create(sd)
sd <- file.path(sd, "NormalDistribution")
if (!file.exists(sd)) dir.create(sd)

contourQuantil <- c(0.75, 0.90, 0.95, 0.99)
x <- seq(from = -3, to = 3, length.out = 500)
contourLevel <- quantile(dnorm(x), probs = contourQuantil)
n <- length(contourLevel)
```

```
contourColor <- rev(colorRampPalette(c("#0000ff",
    "#99ff00"))(n)) # we use American spelling of "colour" as
    functions in R do so

pdf(file = paste0(sd,"/univariateCasePDF.pdf"))
par(mar = c(5, 5, 4, 2) + 0.1)
curve(dnorm(x), from = -3, to = 3,
      xlab = "x", ylab = expression(f["0,1"](x)),
      col = "gray50", lwd = 2, cex.lab = 2)

inverseGenerator = function(f, lower = 0, upper = 100){
  function(y) uniroot((function(x) f(x) - y), lower = lower,
     upper = upper)[1] }
inverseDnorm <- inverseGenerator(function(x) dnorm(x))

contourPoint <- matrix(rep(0, 4*n), ncol = 2)
for (i in 1:(n)){
  contourPoint[i,] <- c(inverseDnorm(contourLevel[i])$root,
     contourLevel[i])
  contourPoint[2*n-i+1,] <- c(-contourPoint[i,1],
     contourPoint[i,2]) }

points(contourPoint[,1], contourPoint[,2], cex = 1.7,
       pch = 21, bg = c(contourColor, rev(contourColor)), col =
          "black")
dev.off()
```

Next we want to draw the contour lines (in the univariate case points) on their own. We do this in a one dimensional plot and encode the contour level with different colours. A small issue was the box of the legend. As a two line title is bad supported, we need to do draw the rectangle by hand. `bty = "n"` removes the line of *legend()* first, and we will add it manually afterwards.

```
sd <- file.path(getwd(), "..", "R-Pictures",
    "NormalDistribution")

pdf(file = paste0(sd,"/univariateCaseContour.pdf"))
plot(contourPoint[,1], rep(0.01, 2*n), ylim = c(0,1), xlim =
    c(-3,3), axes = F,
     ann = FALSE, pch = 21, bg = c(contourColor,
        rev(contourColor)), col = "black")
pos <- legend(x = "right", legend = contourQuantil, title =
    "Contours at\n quantiles",
              cex = 1.7, pch = 21, pt.bg = contourColor, col =
                 "black", bty = "n")
rect(pos$rect$left, pos$rect$top - pos$rect$h, pos$rect$left +
    pos$rect$w, pos$rect$top + 0.1)
```

```
axis(side = 1, at = c(round(contourPoint[,1], 2), 0, -3, 3),
   cex.axis = .8)
dev.off()
```

A.2 The Bivariate Normal Distribution

Now we want to plot the probability density function of a two dimensional standard normal distribution with zero correlation. So right after setting up where to save the file, we compute the values to draw using the *outer* function. For this we have to define the function f externally, which calculates the density of a bivariate standard normal distribution. Now we set up the contour levels and colours similar to the one dimensional case. We choose the distance between points to evaluate as `by = 0.25` in order to get a nice picture (meaning with appropriate portions of black lines and coloured surface). Then we specify the device on which to plot, adjust the parameters for our graphic and use the *persp* function with appropriate chosen ingredients to plot the probability density function. As we want to add the contour lines to this plot, we calculate the inverse of the probability density function similar to above and use the symmetry of a two dimensional standard normal distribution to draw the contour lines. These are circles in our case.

```
sd <- file.path(getwd(), "..", "R-Pictures",
   "NormalDistribution")

f <- function(x,y) dmvnorm(cbind(x,y))
x <- y <-  seq(-3,3, by = 0.25)
z <- outer(x,y,f)

contourQuantil <- c(0.75, 0.90, 0.95, 0.99)
contourLevel <- quantile(z, probs = contourQuantil)
n <- length(contourLevel)
contourColor <- rev(colorRampPalette(c("#0000ff",
   "#99ff00"))(n))

pdf(file = paste0(sd,"/bivariateCasePDF.pdf"))
perspMat <- persp(x,y,z, theta=30, phi=20,
                  xlab=expression(x1), ylab = expression(x2),
                     zlab="", expand = .7,
                  col="gray80", ticktype = "detailed", cex.lab
                     = 1.4)

inverseGenerator = function(f, lower = 0, upper = 3){
  function(y) uniroot((function(x) f(x) - y), lower = lower,
     upper = upper)[1] }
inverseDmvnorm <- inverseGenerator(function(x) dmvnorm(c(x,0)))
```

```
phi <- seq(0, 2*pi, len = 201)
for (p in contourLevel){
  r <- inverseDmvnorm(p)$root # radius of contourLevel p
  rx <- r*cos(phi)
  ry <- r*sin(phi)
  lines(trans3d(rx, ry, f(rx, ry), perspMat), col =
      contourColor[which(contourLevel == p)], lwd=3) }
dev.off()
```

Finally we want to draw the contour lines on their own. We specify where to save the file and open the device. Then we adjust the parameters for our plotting function *contour()*.

```
sd <- file.path(getwd(), "..", "R-Pictures",
    "NormalDistribution")

f <- function(x,y) dmvnorm(cbind(x,y))
x <- y <-  seq(-3,3, by = 0.05)
z <- outer(x,y,f)

pdf(file = paste0(sd,"/bivariateCaseContour.pdf"))
contour(x,y,z, xlim=c(-3,3), ylim=c(-3,3),
        xlab = expression(x[1]), ylab = expression(x[2]),
        levels = contourLevel, labels = round(contourLevel, 2),
        labcex = 1.0, cex.lab = 1.4, col = contourColor, lwd =
           2)
dev.off()
```

Appendix B

R-Code Copula Theory

In this chapter we want to provide code we used for the visualizations in Chapter 2.

B.1 Function for Plotting Contours of Bivariate Copulae

For our function we use the method *BiCopTau2ParX*, which is defined along with the other functions for visualizing trivariate copulae in the next chapter. It is based on methods of the package **VineCopula**.

We want to draw contour plots of bivariate copulae and save these at an appropriate place. The containing folder will be created first if it does not exist so far. We use these bivariate copulae as building blocks later on. For this we provide the following function. We use the graphic parameters, that we use throughout the thesis. During the function it is checked, whether we have to take care about a second parameter (for example attach it to our table) or not. We replace NA's which sometimes happen at the real edges of the plotting area (numerical issue, but are 0). As we want to position our table on the top of the current line in the caption, we have to remove the top rule and insert the "[t]" via *gsub* using regular expressions. We further denote the first parameter by θ_1, the second one by θ_2 and save the resulting table as a `.tex` document. We input this to the caption in LaTeX.

```
sd <- file.path(getwd(), "..", "R-Pictures")
if (!file.exists(sd)) dir.create(sd)
sd <- file.path(sd, "CopulaTheory")
if (!file.exists(sd)) dir.create(sd)

plotContour2d <- function(name, fam, par2 = NA,
                          tau = c(0.8, 0.5, 0.3),
                              contourQuantil = c(0.75, 0.9,
                              0.95, 0.99),
                          z1 = seq(-3, 3, by = 0.1), z2 =
                              seq(-3, 3, by = 0.1),
                          cmap=colorRampPalette(c("#0000ff",
                              "#99ff00"))){
```

```r
sd <- file.path(getwd(), "..", "R-Pictures", "CopulaTheory", fam)
if (!file.exists(sd)) dir.create(sd)

par1 <- rep(0, length(tau))# is filled when going through tau values

pdf(file = paste0(sd,"/plot.pdf"), width = 9, height = 3.4)# to get quadratic plot
par(mfrow = c(1,3), ask = FALSE)
for (i in 1:length(tau)){
  par1[i] <- BiCopTau2ParX(fam, tau[i], par2)
  f <- BiCopMetaContour(family = fam, par = par1[i], par2 = par2, PLOT = FALSE)
  f$z[is.na(f$z)] <- 0
  level <- quantile(f$z, probs = contourQuantil)
  contourColor <- rev(cmap(length(contourQuantil)))

  contour(x = f$x, y = f$y, z = f$z, levels = level,
          label = round(level, 2), col = contourColor)
}
dev.off()

if (is.na(par2)){
  m <- matrix(c(tau, round(par1, 2)),
              nrow = 2, byrow = TRUE)
  rownames(m) <- c("$\\tau$", "$\\theta_1$")
  colnames(m) <- c("left", "mid", "right")
}else{
  m <- matrix(c(tau, round(par1, 2), rep(round(par2, 2), length(tau))),
              nrow = 3, byrow = TRUE)
  rownames(m) <- c("$\\tau$", "$\\theta_1$", "$\\theta_2$")
  colnames(m) <- c("left", "mid", "right")
}

write(
  paste(
    name, " copula \\qquad",
    gsub("(tabular}{+)", "tabular}[t]{",
         print(xtable(m, align = "lccc"),
               sanitize.colnames.function = identity,
               sanitize.text.function=function(x){x}, # for being able to pass math-environment ($)
               booktabs = TRUE,
               hline.after = c(0, nrow(m)),
```

```
                   floating = FALSE,
                   print.results = FALSE),
             perl = TRUE),
       paste0("\\label{cop:", name, "}")
     ),
     file = paste0(sd, "/caption.tex"))
}
```

B.2 Plot Contours of Bivariate Copulae

Now we plot the copulae we use as building blocks in our thesis.

```
plotContour2d("Gaussian", 1)
plotContour2d("Student t", 2, 3)
plotContour2d("Clayton", 3)
plotContour2d("Gumbel", 4)
plotContour2d("Frank", 5)
plotContour2d("Joe", 6)
plotContour2d("BB1", 7, 1.3)
plotContour2d("BB6", 8, 1.3)
plotContour2d("BB7", 9, 0.4)
plotContour2d("BB8", 10, 0.7)
plotContour2d("Tawn type 1", 104, 0.9)
plotContour2d("Tawn type 2", 204, 0.9)
```

Appendix C

R-Code Visualization

Here we want to show how we implemented the visualizations of Chapter 4 in *R*. The code is also available in the self written *R* package **copulaSG**.

C.0 Load Relevant Packages

For the visualization of three dimensional copulae in *R*, we first need the *R* package **VineCopula** which provides functionalities for calculating with copulae. Afterwards we want to visualize our results, for which we use the *R* packages **misc3d** and **rgl**. A great introduction in how to work with **rgl** can be found in Feng and Tierney (2008).

```
# calculating copulae
library("VineCopula")

# plotting
library("misc3d")
library("rgl")

# printing the specifications to LaTeX
library("xtable")
```

C.1 Function for Constructing R-Vine Matrices

VineCopula uses one fundamental object for calculating with copulae, namely the *RVineMatrix* object. This is specified by the matrix components `Matrix, family, par` and `par2`, which we will pass via the following function. We always use the same R-vine structure matrix, so we set this up directly.

```
RVMconstruction = function(fam12, fam23, fam13.2,
                           par12, par23, par13.2,
                           par2_12=0, par2_23=0, par2_13.2=0){
  Matrix <- matrix(c(1, 0, 0,
                     3, 2, 0,
                     2, 3, 3),
```

```
                    nrow = 3, ncol = 3, byrow = TRUE)

  family <- matrix(c(0, 0, 0,
                      fam13.2, 0, 0,
                      fam12, fam23, 0),
                    nrow = 3, ncol = 3, byrow = TRUE)

  par <- matrix(c(0, 0, 0,
                   par13.2, 0, 0,
                   par12, par23, 0),
                 nrow = 3, ncol = 3, byrow = TRUE)

  par2 <- matrix(c(0, 0, 0,
                    par2_13.2, 0, 0,
                    par2_12, par2_23, 0),
                  nrow = 3, ncol = 3, byrow = TRUE)

  return(RVineMatrix(Matrix = Matrix, family = family, par =
     par, par2 =  par2))
}
```

C.2 Function for Calculating Probability Densities

As our aim is to plot contour lines of the probability density function of trivariate copulae, we now provide a function for calculating the probability density. More precise, we want to plot the density on the Z scale and therefore we implement a formula to calculate the following (remember Sklar's Theorem):

$$f_Z(z_1, z_2, z_3) = c(\Phi(z_1), \Phi(z_2), \Phi(z_3)) \cdot f_1(z_1) \cdot f_2(z_2) \cdot f_3(z_3)\,,$$

where $c()$ is the probability density of the corresponding copula.

We use the function *RVinePDF* and add the remaining calculations using vector calculations in order to get fast code. We do this by storing as less intermediary results as possible in order to speed calculations further up.

We return a list containing the function values as a three dimensional array, the values of each single axes and the R-vine matrix. With this information we visualize the data in a significant way and are flexible to change perspectives.

```
f_Z = function(z1, z2, z3, RVM){
  z.matrix <- data.matrix(expand.grid(z1,z2,z3))

  f <- array(RVinePDF(pnorm(z.matrix), RVM =
     RVM)*apply(dnorm(z.matrix), MARGIN = 1, FUN = prod),
     dim=c(length(z1),length(z2), length(z3)))
```

```
  res <- list(f = f, z1 = z1, z2 = z2, z3 = z3, RVM=RVM)
  class(res) <- "f_Z"
  return(res)
}
```

C.3 Functions for Plotting Contour Lines

For plotting the contour lines, we use the function *contour3d*, provided in the package **misc3d**. This method needs the data to plot as a three dimensional array passed by the parameter `data`. In our case we will assign the probability densities. Moreover the parameter `level` encodes the different levels of our contour lines. It is set to `NA` by default, as we want to plot contour levels according to quantiles by default. But if one passes values to `level`, these values are preferred.

The transparency is adjusted via the alpha channel `alo` and `ahi`. For further information about the alpha channel, one can look at Porter and Duff (1984) or Wallace (1981). The colour of the map can be chosen via `cmap`, for which by default we use a self defined colour palette from green to blue. One can also adjust the visualization via the parameter `optplot`, which takes care of the viewpoint and the axes to draw. The colour of the axes is adjusted by `colaxis` and smoothing is done via `smooth`.

The *axes3d* function adds the three axes and *rgl.bringtotop* makes the drawing directly visible and also is needed as on some systems, the snapshot will include content from other windows if they cover the active rgl window. Also we do something like a hack to get equal axes for all scenes. We overrite the parameter `expand` first. This handles the expansion of the axes. Then we add two invisible point at opposite corners and afterwards draw the axes. Now they always include the maximum range of the input.

We want to highlight once again that the function *plotContour* opens an rgl device and in this device one is able to rotate the scenario by hand.

```
plotContour <- function(data, level=NA, contourQuantil=c(0.75,
    0.9, 0.95, 0.99), alo=0.1, ahi=0.5,
    cmap=colorRampPalette(c("#0000ff", "#99ff00")), optplot=1,
    colaxis="gray35", smooth=2, expand=1.03){
  if (is.na(level))
    level = quantile(data$f, probs = contourQuantil)

  n <- length(level)
  contourColor <- rev(cmap(n))
  al <- seq(alo, ahi, len = n)

  if (optplot == 1){# preferred view (cube around plot and
      highlight point of origin)
```

```
    contour3d(data$f, x=data$z1, y=data$z2, z=data$z2,
              level=level, color=contourColor, alpha=al,
                 smooth=smooth)
    points3d(expand*min(data$z1), expand*min(data$z2),
       expand*min(data$z3), alpha=0)
    points3d(expand*max(data$z1), expand*max(data$z2),
       expand*max(data$z3), alpha=0)
    box3d(col = colaxis)
    axes3d(c('x', 'y', 'z'), col = colaxis)
    for (i in c("x", "y", "z")){
      axis3d(i, pos = c(0,0,0), at = c(-1.5, 1.5),
             col = "black", lwd = 2,
             labels = FALSE, tick = FALSE) }
    title3d(xlab = 'z1', ylab = 'z2', zlab = 'z3')
  }else if (optplot == 2){# no cube, axes on the ground
    contour3d(data$f, x=data$z1, y=data$z2, z=data$z2,
              level=level, color=contourColor, alpha=al,
                 smooth=smooth)
    axes3d(c('x', 'y', 'z'), col = colaxis)
    title3d(xlab = 'z1', ylab = 'z2', zlab = 'z3')
  }else if (optplot == 3){# axes through point of origin
    contour3d(data$f, x=data$z1, y=data$z2, z=data$z2,
              level=level, color=contourColor, alpha=al,
                 smooth=smooth)
    axes3d(c('x', 'y', 'z'), pos = c(0,0,0), col = colaxis,
       lwd=2) # thicker lines
    axes3d(c('x', 'y', 'z'), col = colaxis)
    title3d(xlab = 'z1', ylab = 'z2', zlab = 'z3')
    title3d(ylab = 'z1', pos = c(max(data$z1),NA,0)) # looks
       strange, but places z1 correct
    title3d(zlab = 'z2', pos = c(0,max(data$z2),NA))
    title3d(xlab = 'z3', pos = c(NA,0,max(data$z3)))
  }else if (optplot == 11){# standard axes, standard view
    # we interchange parameters in order to get the axes right
       (for contour3d)
    temp <- f_Z(data$z2, data$z3, data$z1, RVM = data$RVM)
    contour3d(temp$f, x=data$z1, y=data$z2, z=data$z2,
              level=level, color=contourColor, alpha=al,
                 smooth=smooth)
    axes3d(c('x', 'y', 'z'), col = colaxis)
    title3d(xlab = 'z2', ylab = 'z3', zlab = 'z1')
  }else{
    stop("Your input for 'optplot' is not supported.")
  }
  rgl.bringtotop()
}
```

To keep things separated (calculating the data and plotting it), we calculate the probability densities first with *f_Z* and pass them to our plotting-function via the `data` argument. But for comfortable and fast generating of the visualizations for the thesis, we provide the function *vis3Dvine*. We automatically save the pictures taken from different angles consecutively named. We can choose our desired points of view by passing to the `angle` parameter a $n \times 2$ matrix containing the θ-angle in the first and the ϕ-angle in the second column. The only way to adjust resolution is via adjusting the plotting area. We set the parameters to a maximum at our current PC, but one can change the size of the plotting window via `windowRect`. If `verbose` is set to true, one will see the summary of the density values (`data$f_Z`).

```
vis3Dvine <- function(data, angle =
   matrix(c(-12,5,20,5,100,5,17,68),
                                                ncol=2, byrow=TRUE)
                          , engine = "rgl", windowRect = c(10, 30,
                            880, 900),
                          verbose = FALSE, ...){
  plotContour(data, ...)
  # we enlarge the plotting area first
  par3d(windowRect = windowRect)

  for(i in 1:nrow(angle)){
    rgl.viewpoint(angle[i,1], angle[i,2])
    rgl.snapshot(paste0(i,".png"))
  }

  if(verbose)
    summary(data$f_Z)
}
```

C.4 Functions for Handling Scenarios

We want to plot lots of different scenarios later on and therefore we store the scenarios in a data frame, which is set up now.

```
scenario <- data.frame(label = character(), name = character(),
                           fam12 = numeric(), fam23 = numeric(),
                              fam13.2 = numeric(),
                           par12 = numeric(), par23 = numeric(),
                              par13.2 = numeric(),
                           par2_12 = numeric(), par2_23 =
                              numeric(), par2_13.2 = numeric(),
                           tau12 = numeric(), tau23 = numeric(),
                              tau13.2 = numeric(),
```

```
                       RVM = list(),
                       stringsAsFactors = FALSE)
```

Now we have got an empty data frame and want to provide a function for adding different scenarios.

As we want our scenarios to be as comparable as possible, we want to print them with same dependence structure (meaning same τ values) and only with changing copula families. For this purpose, we can use *BiCopTau2Par* from the **VineCopula** package. But as this function only works for some copula families (0,1,2,3,4,5,6 and their rotated versions), we need to enlarge its possibilities. So we provide an extended version named *BiCopTau2ParX*, which will be able to calculate the corresponding first parameter given a proper τ value, even if the second parameter also influences the τ. We will fix this issue by letting the user set a second parameter and pass it to our function. Then the first parameter is calculated, such that this and the passed by second parameter correspond to the given τ value. In the end our function expands the *BiCopTau2Par* function with the copula families $7, 8, 9, 10, 104, 204$.

```
BiCopTau2ParX <- function(family, tau, par2=NA){
  if (length(family) != 1 || length(tau) != 1 || length(par2)
     != 1)
    stop("All your inputs have to be scalars.")
  if (!(family %in% c(0, 1, 2, 3, 4, 5, 6, 13, 14, 16, 23,
                      24, 26, 33, 34, 36, 41, 51, 61, 71,
                      7, 8, 9, 10, 104, 204)))
    stop("Copula family not implemented.")
  if (tau < -1 || tau > 1)
    stop("Tau value has to be at least between -1 and 1.")

  par1 <- NA # initialize value

  if (family %in% c(0, 1, 3, 4, 5, 6, 13, 14, 16, 23,
                    24, 26, 33, 34, 36, 41, 51, 61, 71)){
    if (length(par2[!is.na(par2)]) > 0)
      warning("Your input for par2 was ignored.")
    par1 <- BiCopTau2Par(family, tau)
  }
  else if (is.na(par2)){# else if important!
    stop("You have to pass par2")
  }
  else if(family == 2){
    if (par2 < 2)
      stop("t: par2 has to be >= 2")
    par1 <- BiCopTau2Par(family, tau)
  }
  else if (family == 7){
```

```
    if (tau <= 0)
      stop("BB1 copula cannot be used for tau<=0.")
    else if (par2 < 1)
      stop("BB1: par2 has to be >= 1")
    par1 <- (2/(par2*(1-tau)) - 2)
  }
  else if (family == 8){
    if (tau < 0)
      stop("BB6 copula cannot be used for tau < 0.")
    else if (par2 < 1)
      stop("BB6: par2 has to be >= 1")
    inverseGenerator <- function(f, lower = 1, upper = 100){
      function(y) uniroot((function(x) f(x) - y), lower =
         lower, upper = upper)[1] }
    inverseFun <- inverseGenerator(function(par1)
       BiCopPar2Tau(family = family, par1, par2 = par2))
    tryCatch(
      par1 <- inverseFun(tau)$root,
      error = function(e) {
        stop(paste("Error occurred: Probably one can't fulfill
           your tau-value with given your parameter two.",
                  "\n\t original message: \n\t",
                  e$message, "\n"))
      }
    )
  }
  else if (family == 9){
    if (tau <= 0)
      stop("BB7 copula cannot be used for tau<=0.")
    else if (par2 <= 0)
      stop("BB7: par2 has to be > 0")
    inverseGenerator <- function(f, lower = 1, upper = 100){
      function(y) uniroot((function(x) f(x) - y), lower =
         lower, upper = upper)[1] }
    inverseFun <- inverseGenerator(function(par1)
       BiCopPar2Tau(family = family, par1, par2 = par2))
    tryCatch(
      par1 <- inverseFun(tau)$root,
      error = function(e) {
        stop(paste("Error occurred: Probably one can't fulfill
           your tau-value with given your parameter two.",
                   "\n\t original message: \n\t",
                   e$message, "\n"))
      }
    )
  }
```

```r
else if(family == 10){
  if (tau <= 0)
    stop("BB8 copula cannot be used for tau<=0.")
  else if (par2 <= 0 || par2 > 1)
    stop("BB8: par2 has to be > 0 and <= 1")
  inverseGenerator <- function(f, lower = 1, upper = 100){
    function(y) uniroot((function(x) f(x) - y), lower =
        lower, upper = upper)[1] }
  inverseFun <- inverseGenerator(function(par1)
     BiCopPar2Tau(family = family, par1, par2 = par2))
  tryCatch(
    par1 <- inverseFun(tau)$root,
    error = function(e) {
      stop(paste("Error occurred: Probably one can't fulfill
          your tau-value with given your parameter two.",
                 "\n\t original message: \n\t",
                 e$message, "\n"))
    }
  )
}
else if (family == 104){
  if (tau < 0)
    stop("Tawn type 1 copula cannot be used for tau<0.")
  else if (par2 <= 0 || par2 > 1)
    stop("Tawn type 1: par2 has to be > 0 and <= 1")
  inverseGenerator <- function(f, lower = 1, upper = 100){
    function(y) uniroot((function(x) f(x) - y), lower =
        lower, upper = upper)[1] }
  inverseFun <- inverseGenerator(function(par1)
     BiCopPar2Tau(family = family, par1, par2 = par2))
  tryCatch(
    par1 <- inverseFun(tau)$root,
    error = function(e) {
      stop(paste("Error occurred: Probably one can't fulfill
          your tau-value with given your parameter two.",
                 "\n\t original message: \n\t",
                 e$message, "\n"))
    }
  )
}
else if (family == 204){
  if (tau < 0)
    stop("Tawn type 2 copula cannot be used for tau<0.")
  else if (par2 <= 0 || par2 > 1)
    stop("Tawn type 2: par2 has to be > 0 and <= 1")
  inverseGenerator <- function(f, lower = 1, upper = 100){
```

```
      function(y) uniroot((function(x) f(x) - y), lower =
          lower, upper = upper)[1] }
    inverseFun <- inverseGenerator(function(par1)
       BiCopPar2Tau(family = family, par1, par2 = par2))
    tryCatch(
      par1 <- inverseFun(tau)$root,
      error = function(e) {
        stop(paste("Error occurred: Probably one can't fulfill
           your tau-value with given your parameter two.",
                   "\n\t original message: \n\t",
                   e$message, "\n"))
      }
    )
  }

  return(par1)
}
```

So now it is time to provide a way for adding scenarios to our data frame. This is done by the following function *addScenario* which takes the copula families, the τ values and the second parameters for each dimension. The corresponding first parameter is then calculated via *BiCopTau2ParX*. To be able to add also scenarios directly with parameter values, we provide the function *addScenarioPar*. We assign a name to each scenario, where ‘ ’ denotes tau values and ‘-’ denotes par values. But this notation is not important, as later on the table with the whole configuration is printed. We just use the name of the scenario for putting a `.txt` document with the name of the scenario into the folder of the corresponding pictures. As we want to simplify things, we round values to two digits, which makes it easy to describe scenarios.

```
addScenario <- function(label, fam12, fam23, fam13.2,
                        tau12=0.8, tau23=0.5, tau13.2=0.3,
                        par2_12=NA, par2_23=NA, par2_13.2=NA){
  n <- nrow(scenario) + 1
  scenario[n,1] <- label
  scenario[n,2] <- paste0(BiCopName(fam12), BiCopName(fam23),
     BiCopName(fam13.2), " ", round(tau12, 2), " ",
     round(tau23, 2), " ", round(tau13.2, 2))
  scenario[n,3] <- fam12
  scenario[n,4] <- fam23
  scenario[n,5] <- fam13.2
  scenario[n,6] <- round(BiCopTau2ParX(fam12, tau12, par2_12),
     2)
  scenario[n,7] <- round(BiCopTau2ParX(fam23, tau23, par2_23),
     2)
  scenario[n,8] <- round(BiCopTau2ParX(fam13.2, tau13.2,
     par2_13.2), 2)
```

```
  scenario[n,9] <- round(par2_12, 2)
  scenario[n,10] <- round(par2_23, 2)
  scenario[n,11] <- round(par2_13.2, 2)
  scenario[n,12] <- round(tau12, 2)
  scenario[n,13] <- round(tau23, 2)
  scenario[n,14] <- round(tau13.2, 2)
  return(scenario)
}

addScenarioPar <- function(label, fam12, fam23, fam13.2,
                           par12, par23, par13.2,
                           par2_12 = 0, par2_23 = 0, par2_13.2
                              = 0){
  n <- nrow(scenario) + 1
  scenario[n,1] <- label
  scenario[n,2] <- paste0(BiCopName(fam12), BiCopName(fam23),
     BiCopName(fam13.2), "-", par12, "-", par23, "-", par13.2)
  scenario[n,3] <- fam12
  scenario[n,4] <- fam23
  scenario[n,5] <- fam13.2
  scenario[n,6] <- round(par12, 2)
  scenario[n,7] <- round(par23, 2)
  scenario[n,8] <- round(par13.2, 2)
  scenario[n,9] <- round(par2_12, 2)
  scenario[n,10] <- round(par2_23, 2)
  scenario[n,11] <- round(par2_13.2, 2)
  scenario[n,12] <- round(BiCopPar2Tau(fam12, par12, par2_12),
     2)
  scenario[n,13] <- round(BiCopPar2Tau(fam23, par23, par2_23),
     2)
  scenario[n,14] <- round(BiCopPar2Tau(fam13.2, par13.2,
     par2_13.2), 2)
  return(scenario)
}
```

Before setting the scenarios up, we need a function that is able to print the specification of the actual scenario. The function *scenarioToLatex* takes all the relevant information about the scenario (copula family, parameter one and two, τ value), puts them together in a matrix and exports this matrix as a table to a `.tex` file using the function *xtable*.

```
# include.rownames is not working directly in xtable()
# -> need print() to make include.rownames active
# -> need write() to save things to the disk
scenarioToLatex <- function(fam12, fam23, fam13.2,
                            par12, par23, par13.2,
                            par2_12, par2_23, par2_13.2,
```

```
                              tau12, tau23, tau13.2,
                              label){
  m <- matrix(c("$12$", BiCopName(fam12), tau12, par12, par2_12,
                "$23$", BiCopName(fam23), tau23, par23, par2_23,
                "$13|2$", BiCopName(fam13.2), tau13.2, par13.2,
                   par2_13.2),
              ncol = 5, byrow = TRUE,
              dimnames = list(NULL, c("Pair Copula", "Family",
                "$\\tau$", "$\\theta_1$", "$\\theta_2$")))
  options(warn = -1) # there is a warning when passing C{} to
     align, we do this in order to define a new column type
     with fixed width but centred
  write(
    gsub("(tabular}{+)", "tabular}[t]{",
         print(xtable(m, align =
            "lcC{1.5cm}C{1.2cm}C{1.2cm}C{1.2cm}"), # pass
            ncol()+1 arguments (rowname)
               include.rownames = FALSE, booktabs = TRUE,
                  floating = FALSE,
               hline.after = c(0, nrow(m)),
               sanitize.text.function=function(x){x}, # for
                  being able to pass math-environment ($)
               print.results = FALSE
         ),
         perl = TRUE),
    file = "00configuration.tex")
}
```

To handle all scenarios and create our files, we now define the function *scenarioToFile*, which takes the number of the scenario to visualize and the data frame. It stores the parameters of the scenario in meaningful named variables first and then sets the working directory to a folder where to save the results of the visualization. We adjust the quality of our plot via the points of evaluation with the `by=` argument and afterwards action is taken. At the very end we produce a table containing all scenarios. But this is highly specific for our thesis and is only a complete table after the visualization of uranium has been run, so we set the parameter `table` to FALSE by default.

```
scenarioToFile <- function(num, scenarioIn, table = FALSE){
  if (num < 1 || num > nrow(scenario))
    stop("Your number has to between 1 and the number of rows
       of scenario.")

### get the variables
  scenarioIn[num,1] -> label
  scenarioIn[num,2] -> name
  scenarioIn[num,3] -> fam12
```

```
  scenarioIn[num,4] -> fam23
  scenarioIn[num,5] -> fam13.2
  scenarioIn[num,6] -> par12
  scenarioIn[num,7] -> par23
  scenarioIn[num,8] -> par13.2
  scenarioIn[num,9] -> par2_12
  scenarioIn[num,10] -> par2_23
  scenarioIn[num,11] -> par2_13.2
  scenarioIn[num,12] -> tau12
  scenarioIn[num,13] -> tau23
  scenarioIn[num,14] -> tau13.2

### set the working directory
  wd <- getwd()
  on.exit(setwd(wd)) # reset working directory at the end
  sd <- file.path(wd, "..", "R-Pictures")
  if (!file.exists(sd)) dir.create(sd)
  sd <- file.path(sd, "Scenarios")
  if (!file.exists(sd)) dir.create(sd)

  folder <- file.path(sd, label)
  if (!file.exists(folder)) dir.create(folder)
  setwd(folder)

### do the action (calculate, images, latex-table)
  z1 <- z2 <- z3 <- seq(-3, 3, by=0.05)
  RVM <- RVMconstruction(fam12, fam23, fam13.2,
                         par12, par23, par13.2,
                         par2_12, par2_23, par2_13.2)
  f_RVM <- f_Z(z1, z2, z3, RVM)

  vis3Dvine(f_RVM)

  if (is.na(par2_12)) par2_12 <- 0
  if (is.na(par2_23)) par2_23 <- 0
  if (is.na(par2_13.2)) par2_13.2 <- 0

  scenarioToLatex(fam12, fam23, fam13.2,
                  par12, par23, par13.2,
                  par2_12, par2_23, par2_13.2,
                  tau12, tau23, tau13.2,
                  label)

  # put label in a tex file
  write(paste0("\\label{", label, "}"), file = "00label.tex")
```

```
  # put one text file with the name in the folder
  write("", file = paste0(name, ".txt"))

### produce the table containing all scenarios
  if(table){
    if (num == 1){
      write("\\section{Table of Scenarios}",
            file = file.path(sd, "ScenarioTable.tex"))
      write("{\\footnotesize",
            file = file.path(sd, "ScenarioTable.tex"), append =
               TRUE)

      write("\\begin{longtable}{r  r >{$}c<{$} >{$}c<{$}
         >{$}c<{$}  r >{$}c<{$} >{$}c<{$} >{$}c<{$}  r
         >{$}c<{$} >{$}c<{$} >{$}c<{$}}",
            file = file.path(sd, "ScenarioTable.tex"), append =
               TRUE )

      write("& \\multicolumn{4}{c}{Copula $c_{12}$} &
         \\multicolumn{4}{c}{Copula $c_{23}$}  &
         \\multicolumn{4}{c}{Copula $c_{13;2}$}\\\\",
            file = file.path(sd, "ScenarioTable.tex"), append =
               TRUE )
      write("\\cmidrule(lr){2-5} \\cmidrule(lr){6-9}
         \\cmidrule(lr){10-13}",
            file = file.path(sd, "ScenarioTable.tex"), append =
               TRUE )
      write("p. & Fam & \\tau & \\theta_1 & \\theta_2  & Fam &
         \\tau & \\theta_1 & \\theta_2  & Fam & \\tau &
         \\theta_1 & \\theta_2 \\\\",
            file = file.path(sd, "ScenarioTable.tex"), append =
               TRUE )
      write("\\midrule \\endfirsthead",
            file = file.path(sd, "ScenarioTable.tex"), append =
               TRUE )

      write("& \\multicolumn{4}{c}{Copula $c_{12}$} &
         \\multicolumn{4}{c}{Copula $c_{23}$}  &
         \\multicolumn{4}{c}{Copula $c_{13;2}$}\\\\",
            file = file.path(sd, "ScenarioTable.tex"), append =
               TRUE )
      write("\\cmidrule(lr){2-5} \\cmidrule(lr){6-9}
         \\cmidrule(lr){10-13}",
            file = file.path(sd, "ScenarioTable.tex"), append =
               TRUE )
```

```
      write("p. & Fam & \\tau & \\theta_1 & \\theta_2  & Fam &
         \\tau & \\theta_1 & \\theta_2  & Fam & \\tau &
         \\theta_1 & \\theta_2 \\\\",
            file = file.path(sd, "ScenarioTable.tex"), append =
               TRUE )
      write("\\midrule \\endhead",
            file = file.path(sd, "ScenarioTable.tex"), append =
               TRUE )

      write("\\bottomrule \\endlastfoot",
            file = file.path(sd, "ScenarioTable.tex"), append =
               TRUE )

      write(
        paste(
          paste0("\\pageref{", label, "}"),
          BiCopName(fam12), tau12, par12, par2_12,
          BiCopName(fam23), tau23, par23, par2_23,
          BiCopName(fam13.2), tau13.2, par13.2,
             paste0(par2_13.2, "\\\\")
          , sep = " & "
        ), file = file.path(sd, "ScenarioTable.tex"), append =
           TRUE )
    }
    else if (num == 48){
      write(
        paste(
          "NA",
          BiCopName(fam12), tau12, par12, par2_12,
          BiCopName(fam23), tau23, par23, par2_23,
          BiCopName(fam13.2), tau13.2, par13.2,
             paste0(par2_13.2, "\\\\")
          , sep = " & "
        ), file = file.path(sd, "ScenarioTable.tex"), append =
           TRUE )
      }
    else if (num > 1 && num <= nrow(scenario)){
      write(
        paste(
          paste0("\\pageref{", label, "}"),
          BiCopName(fam12), tau12, par12, par2_12,
          BiCopName(fam23), tau23, par23, par2_23,
          BiCopName(fam13.2), tau13.2, par13.2,
             paste0(par2_13.2, "\\\\")
          , sep = " & "
```

```
        ), file = file.path(sd, "ScenarioTable.tex"), append =
            TRUE )
      }
  }
}
```

C.5 Include Scenarios

Now we are ready to add lots of scenarios to our data frame.

For internal usage, we separate the label with '+', not with ':' as the following characters are not allowed to be used in folder names: '<' '>' ':' '"' '\' '/' '|' '*' '?'. Also LaTeX has forbidden characters, for example '#'. We use the '-' when we summarize more visualizations in one figure in LaTeX.

```
# Trivariate Gaussian Copula
scenario <- addScenario("vis+Gauss1", 1, 1, 1)
scenario <- addScenario("vis+Gauss2", 1, 1, 1,
                        0.3, 0.5, 0.8)

# Exchangeable Copulae
par12 <- par23 <- rho <- .95
par13.2 = rho*(1-rho)/(1-rho^2)
scenario <- addScenarioPar("vis+GaussEx-1", 1, 1, 1,
                           par12, par23, par13.2)

par12 <- par23 <- rho <- .71
par13.2 = rho*(1-rho)/(1-rho^2)
scenario <- addScenarioPar("vis+GaussEx-2", 1, 1, 1,
                           par12, par23, par13.2)

par12 <- par23 <- rho <- .45
par13.2 = rho*(1-rho)/(1-rho^2)
scenario <- addScenarioPar("vis+GaussEx-3", 1, 1, 1,
                           par12, par23, par13.2)

# AR(1) copulae
scenario <- addScenarioPar("vis+GaussAR-1", 1, 1, 1,
                           0.95, 0.95, 0)
scenario <- addScenarioPar("vis+GaussAR-2", 1, 1, 1,
                           0.71, 0.71, 0)
scenario <- addScenarioPar("vis+GaussAR-3", 1, 1, 1,
                           0.45, 0.45, 0)
```

```r
# Student t Copulae
scenario <- addScenario("vis+T1-1", 2, 2, 2,
                        0, 0, 0,
                        3, 3, 4)
scenario <- addScenario("vis+T1-2", 2, 2, 2,
                        0, 0, 0,
                        6, 6, 7)
scenario <- addScenario("vis+T1-3", 2, 2, 2,
                        0, 0, 0,
                        25, 25, 26)

scenario <- addScenario("vis+T2-1", 2, 2, 2,
                        0.8, 0.5, 0.3,
                        3, 3, 4)
scenario <- addScenario("vis+T2-2", 2, 2, 2,
                        0.8, 0.5, 0.3,
                        6, 6, 7)
scenario <- addScenario("vis+T2-3", 2, 2, 2,
                        0.8, 0.5, 0.3,
                        25, 25, 26)

# Independence Copulae
scenario <- addScenario("vis+Ind-1", 0, 0, 0,
                        0, 0, 0)
scenario <- addScenario("vis+Ind-2", 1, 1, 1,
                        0, 0, 0)
scenario <- addScenario("vis+Ind-3", 1, 2, 3,
                        0, 0, 0.01,
                        NA, 3, NA)



# Scenarios out of Archimedean Copulae with one Parameter
scenario <- addScenario("vis+CGFJ1", 3, 3, 3)
scenario <- addScenario("vis+CGFJ2", 4, 4, 4)
scenario <- addScenario("vis+CGFJ3", 5, 5, 5)
scenario <- addScenario("vis+CGFJ4", 6, 6, 6)

scenario <- addScenario("vis+CGFJ5", 3, 4, 5)
scenario <- addScenario("vis+CGFJ6", 5, 4, 3)
scenario <- addScenario("vis+CGFJ7", 6, 5, 4)
scenario <- addScenario("vis+CGFJ8", 4, 6, 3)



# Scenarios out of Archimedean Copulae with two Parameters
```

```
scenario <- addScenario("vis+BBBBTT1", 7, 7, 7,
                        .8, .5, .3,
                        1.3, 1.3, 1.3)
scenario <- addScenario("vis+BBBBTT2", 8, 8, 8,
                        .8, .5, .3,
                        1.5, 1.3, 1.3)
scenario <- addScenario("vis+BBBBTT3", 9, 9, 9,
                        .8, .5, .3,
                        .4, .4, .4)

scenario <- addScenario("vis+BBBBTT4", 10, 10, 10,
                        .8, .5, .3,
                        .7, .7, .7)
scenario <- addScenario("vis+BBBBTT5", 104, 104, 104,
                        .8, .5, .3,
                        .9, .9, .9)
scenario <- addScenario("vis+BBBBTT6", 204, 204, 204,
                        .8, .5, .3,
                        .9, .9, .9)

scenario <- addScenario("vis+BBBBTT10", 7, 8, 9,
                        .8, .5, .3,
                        1.3, 1.3, .4)
scenario <- addScenario("vis+BBBBTT11", 10, 104, 204,
                        .8, .5, .3,
                        .7, .9, .9)
scenario <- addScenario("vis+BBBBTT12", 104, 7, 10,
                        .8, .5, .3,
                        .9, 1.3, .7)
scenario <- addScenario("vis+BBBBTT13", 204, 8, 9,
                        .8, .5, .3,
                        .9, 1.3, .4)



# Scenarios with no Restriction of Copula Families
scenario <- addScenario("vis+NRCF1", 5, 9, 3,
                        .8, .5, .3,
                        NA, .4, NA)
scenario <- addScenario("vis+NRCF2", 7, 2, 6,
                        .8, .5, .3,
                        1.3, 3, NA)
scenario <- addScenario("vis+NRCF3", 3, 1, 4)
scenario <- addScenario("vis+NRCF4", 104, 2, 4,
                        .8, .5, .3,
                        .9, 3, NA)
```

```
scenario <- addScenario("vis+NRCF5", 5, 3, 8,
                        .8, .5, .3,
                        NA, NA, 1.3)



# Scenarios with no Restriction on tau Values
scenario <- addScenario("vis+NRT1", 3, 4, 5,
                        .3, .2, .8)
scenario <- addScenario("vis+NRT2", 2, 3, 1,
                        0, .33, 0,
                        3, NA, NA)
scenario <- addScenario("vis+NRT3", 2, 4, 2,
                        0, .5, 0,
                        3, NA, 5)
scenario <- addScenario("vis+NRT4", 1, 3, 7,
                        0, .33, .75,
                        NA, NA, 2)
scenario <- addScenario("vis+NRT5", 4, 2, 9,
                        .5, 0, .6,
                        NA, 3, 2)



# Scenarios in \cite{Mai.2012}
scenario <- addScenario("vis+Mai1", 4, 23, 3,
                        .8, -.54, .26)
scenario <- addScenario("vis+Mai2", 2, 4, 2,
                        .59, .43, -.80,
                        2.1, NA, 2.5)
scenario <- addScenario("vis+Mai3", 16, 6, 13,
                        .35, .92, .91)
scenario <- addScenario("vis+Mai4", 5, 3, 5,
                        -.89, .91, .89)
```

C.6 Visualize Scenarios

The last thing to do is to run with a for-loop through our data frame and visualize every scenario via calling *scenarioToFile*, which is done in one line.

```
for (i in 1:nrow(scenario)) scenarioToFile(i, scenario, table =
   TRUE)
```

Appendix D

Overviews

D.1 Copula Families

Family nr.	Copula family	Abbreviation	θ_1	θ_2
0	Independence	I	$\emptyset$	$\emptyset$
1	Gaussian	N	$(-1,1)$	$\emptyset$
2	Student t	t	$(-1,1)$	$(2,\infty)$
3	Clayton	C	$(0,\infty)$	$\emptyset$
4	Gumbel	G	$[1,\infty)$	$\emptyset$
5	Frank	F	$\mathbb{R}\backslash\{0\}$	$\emptyset$
6	Joe	J	$(1,\infty)$	$\emptyset$
7	Clayton-Gumbel	BB1	$(0,\infty)$	$[1,\infty)$
8	Joe-Gumbel	BB6	$[1,\infty)$	$[1,\infty)$
9	Joe-Clayton	BB7	$[1,\infty)$	$(0,\infty)$
10	Joe-Frank	BB8	$[1,\infty)$	$(0,1]$
104	Tawn type 1	Tawn	$[1,\infty)$	$(0,1]$
204	Tawn type 2	Tawn2	$[1,\infty)$	$(0,1]$

D.2 Table of Scenarios

	Copula c_{12}				Copula c_{23}				Copula $c_{13;2}$			
p.	Fam	τ	θ_1	θ_2	Fam	τ	θ_1	θ_2	Fam	τ	θ_1	θ_2
36	N	0.8	0.95	0	N	0.5	0.71	0	N	0.3	0.45	0
37	N	0.3	0.45	0	N	0.5	0.71	0	N	0.8	0.95	0
40	N	0.8	0.95	0	N	0.8	0.95	0	N	0.32	0.49	0
40	N	0.5	0.71	0	N	0.5	0.71	0	N	0.27	0.42	0
40	N	0.3	0.45	0	N	0.3	0.45	0	N	0.2	0.31	0
41	N	0.8	0.95	0	N	0.8	0.95	0	N	0	0	0
41	N	0.5	0.71	0	N	0.5	0.71	0	N	0	0	0
41	N	0.3	0.45	0	N	0.3	0.45	0	N	0	0	0
44	t	0	0	3	t	0	0	3	t	0	0	4
44	t	0	0	6	t	0	0	6	t	0	0	7
44	t	0	0	25	t	0	0	25	t	0	0	26
45	t	0.8	0.95	3	t	0.5	0.71	3	t	0.3	0.45	4
45	t	0.8	0.95	6	t	0.5	0.71	6	t	0.3	0.45	7
45	t	0.8	0.95	25	t	0.5	0.71	25	t	0.3	0.45	26
47	I	0	0	0	I	0	0	0	I	0	0	0
47	N	0	0	0	N	0	0	0	N	0	0	0
47	N	0	0	0	t	0	0	3	C	0.01	0.02	0
49	C	0.8	8	0	C	0.5	2	0	C	0.3	0.86	0
50	G	0.8	5	0	G	0.5	2	0	G	0.3	1.43	0
51	F	0.8	18.19	0	F	0.5	5.74	0	F	0.3	2.92	0
52	J	0.8	8.77	0	J	0.5	2.86	0	J	0.3	1.77	0
54	C	0.8	8	0	G	0.5	2	0	F	0.3	2.92	0
55	F	0.8	18.19	0	G	0.5	2	0	C	0.3	0.86	0
56	J	0.8	8.77	0	F	0.5	5.74	0	G	0.3	1.43	0
57	G	0.8	5	0	J	0.5	2.86	0	C	0.3	0.86	0
59	BB1	0.8	5.69	1.3	BB1	0.5	1.08	1.3	BB1	0.3	0.2	1.3
60	BB6	0.8	5.46	1.5	BB6	0.5	1.98	1.3	BB6	0.3	1.17	1.3
61	BB7	0.8	8.43	0.4	BB7	0.5	2.5	0.4	BB7	0.3	1.42	0.4
62	BB8	0.8	16.91	0.7	BB8	0.5	5.58	0.7	BB8	0.3	3.2	0.7
63	Tawn	0.8	8.28	0.9	Tawn	0.5	2.16	0.9	Tawn	0.3	1.47	0.9
64	Tawn2	0.8	8.28	0.9	Tawn2	0.5	2.16	0.9	Tawn2	0.3	1.47	0.9
66	BB1	0.8	5.69	1.3	BB6	0.5	1.98	1.3	BB7	0.3	1.42	0.4
67	BB8	0.8	16.91	0.7	Tawn	0.5	2.16	0.9	Tawn2	0.3	1.47	0.9
68	Tawn	0.8	8.28	0.9	BB1	0.5	1.08	1.3	BB8	0.3	3.2	0.7
69	Tawn2	0.8	8.28	0.9	BB6	0.5	1.98	1.3	BB7	0.3	1.42	0.4
71	F	0.8	18.19	0	BB7	0.5	2.5	0.4	C	0.3	0.86	0
72	BB1	0.8	5.69	1.3	t	0.5	0.71	3	J	0.3	1.77	0
73	C	0.8	8	0	N	0.5	0.71	0	G	0.3	1.43	0
74	Tawn	0.8	8.28	0.9	t	0.5	0.71	3	G	0.3	1.43	0
75	F	0.8	18.19	0	C	0.5	2	0	BB6	0.3	1.17	1.3
77	C	0.3	0.86	0	G	0.2	1.25	0	F	0.8	18.19	0
78	t	0	0	3	C	0.33	0.99	0	N	0	0	0
79	t	0	0	3	G	0.5	2	0	t	0	0	5
80	N	0	0	0	C	0.33	0.99	0	BB1	0.75	2	2
81	G	0.5	2	0	t	0	0	3	BB7	0.6	2.22	2

	Copula c_{12}				Copula c_{23}				Copula $c_{13;2}$			
p.	Fam	τ	θ_1	θ_2	Fam	τ	θ_1	θ_2	Fam	τ	θ_1	θ_2
83	G	0.8	5	0	C90	−0.54	−2.35	0	C	0.26	0.7	0
84	t	0.59	0.8	2.1	G	0.43	1.75	0	t	−0.8	−0.95	2.5
NA	SJ	0.35	1.98	0	J	0.92	23.73	0	SC	0.91	20.22	0
85	F	−0.89	−34.64	0	C	0.91	20.22	0	F	0.89	34.64	0
90	t	0.53	0.74	8.02	t	0.43	0.62	5.93	t	0.08	0.13	5.65

The "NA" between page 84 and 85 corresponds to the third scenario of Mai and Scherer (2012) which causes strong numerical problems and thus is not shown in the thesis.

Bibliography

Aas, K., Czado, C., Frigessi, A., and Bakken, H. (2009). Pair-copula constructions of multiple dependence. *Insurance: Mathematics and Economics*, 44(2):182–198.

Bedford, T. and Cooke, R. M. (2002). Vines–a new graphical model for dependent random variables. *The Annals of Statistics*, 30(4):1031–1068.

Bedford, T. J. and Cooke, R., editors (2001). *Monte Carlo simulation of vine dependent random variables for applications in uncertainty analysis.*

Brechmann, E. C. and Schepsmeier, U. (2013). Modeling dependence with c- and d-vine copulas: The r package cdvine. *Journal of Statistical Software*, 52.

Czado, C. (2013). Vine copulas with application.

Demarta, S. and McNeil, A. J. (2005). The t copula and related copulas. *International statistical review*, 73(1):111–129.

Eschenburg, P. (01.01.2013). *Properties of extreme-value copulas.* PhD thesis, Technische Universität München.

Feng, D. and Tierney, L. (2008). Computing and displaying isosurfaces in r. *Journal of Statistical Software*, 28(1):1–24.

Frank, M. J. (1979). On the simultaneous associativity of f(x, y) and x+y−f(x, y). *Aequationes Mathematicae*, 19(1):194–226.

Hogg, R. V., McKean, J. W., and Craig, A. T. (2013). *Introduction to mathematical statistics.* Pearson, Boston, 7th ed. edition.

Joe, H. (1996). Families of m-variate distributions with given margins and $m(m-1)/2$ bivariate dependence parameters. In Rüschendorf, L., Schweizer, B., and Taylor, M. D., editors, *Distributions with fixed marginals and related topics*, Institute of Mathematical Statistics Lecture Notes - Monograph Series, pages 120–141. Institute of Mathematical Statistics, Hayward, CA.

Joe, H. (1997). *Multivariate models and dependence concepts*, volume 73 of *Monographs on statistics and applied probability.* Chapman & Hall, London and New York.

Kurowicka, D. and Cooke, R. (2006). *Uncertainty analysis: Mathematical foundations and applications*. John Wiley, Chichester.

Lindskog, F., McNeil, A., and Schmock, U. (2002). Kendall's tau for elliptical distributions.

Mai, J.-F. and Scherer, M. (2012). *Simulating copulas: Stochastic models, sampling algorithms and applications*, volume v. 4 of *Series in quantitative finance*. World Scientific, Singapore and Hackensack, NJ.

Nelsen, R. B. (2006). *An Introduction to Copulas: With 116 examples and 167 exercises (Springer Series in Statistics)*. Springer-Verlag, s.l., 2. aufl. edition.

Porter, T. and Duff, T. (1984). Compositing digital images. In Christiansen, H., Luther, D. A., Mueller, R. M., Weinberg, R. A., and Ellis, R. A., editors, *the 11th annual conference*, pages 253–259.

Wallace, B. A. (1981). Merging and transformation of raster images for cartoon animation. In Green, D., Lucido, T., and Fuchs, H., editors, *the 8th annual conference*, pages 253–262.