Eindhoven University of Technology

BACHELOR

Checkerboard copulas with maximum entropy

literature overview and a case study

Zwerus, Daan

Award date:
2021

Link to publication

Disclaimer
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 20. Apr. 2025

Technische Universiteit
Eindhoven
University of Technology

Department of mathematics
Statistics
www.tue.nl

Author
Daan Zwerus (1018697)

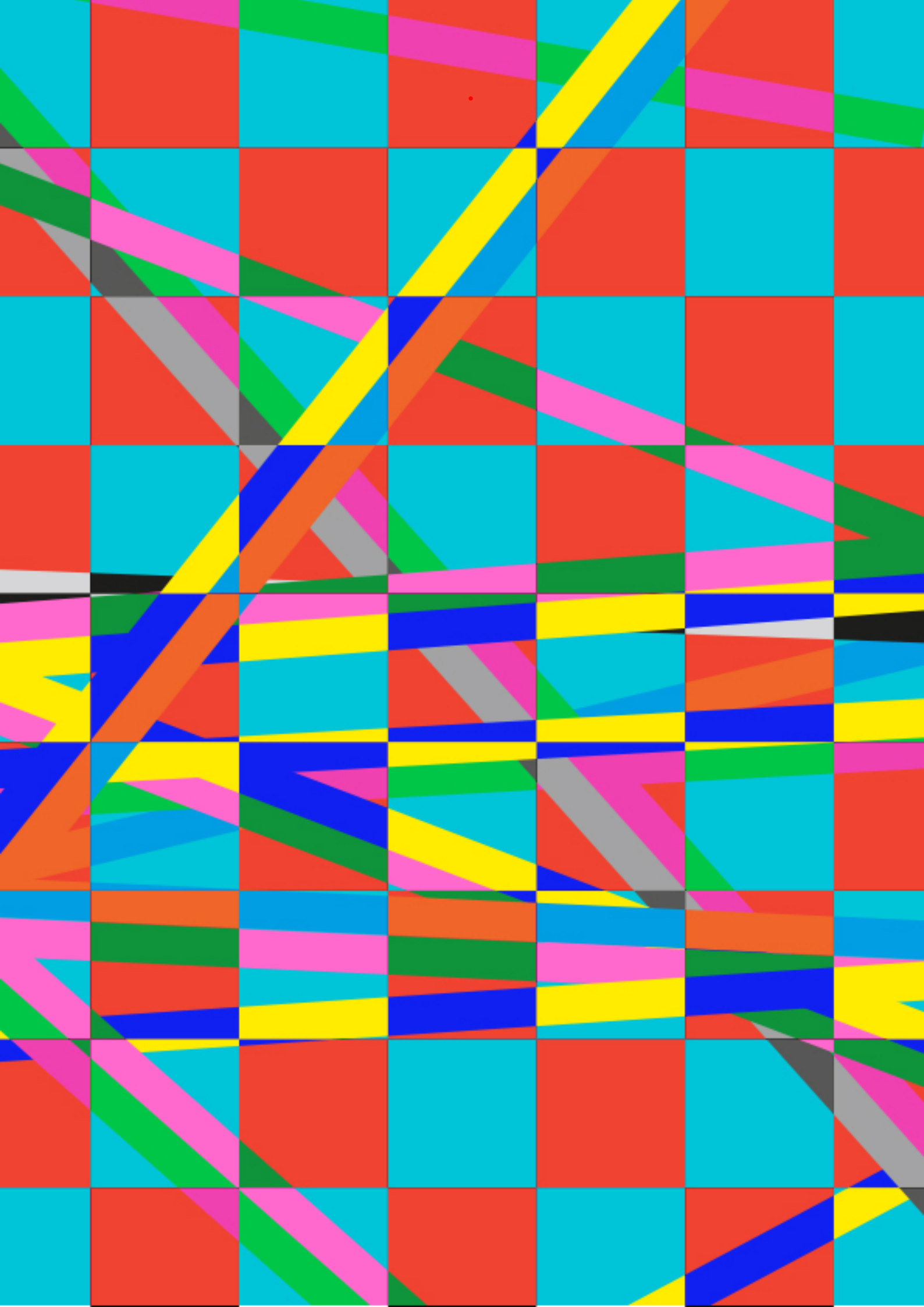
Supervisor
Dr. Perrone

Date
August 30, 2021

Checkerboard copulas with maximum entropy: literature overview and a case study

Daan Zwerus (1018697)
d.a.j.j.zwerus@student.tue.nl

Where innovation starts



Abstract

Copulas are mathematical tools to describe the dependence between random variables. Copulas can be defined as multivariate cumulative distribution functions with marginal distributions that are uniform on $[0, 1]$. Such functions are useful in numerous types of applications, for example, they can be used in hydrology or finance. In this thesis, we focus on a special class of copulas called checkerboard copulas. We create a checkerboard copula by maximizing the amount of disorder, namely entropy. This results in a compromise between dependence and independence. Checkerboard copulas enable us to create simple analytic models useful to generate synthetic data when only partial information is available. In this thesis, we review the theory of checkerboard copulas, and we analyze a case study on monthly rainfall totals in Eindhoven.
***Keywords*—** copulas, checkerboard copulas, dependence, hydrology

Technische Universiteit
Eindhoven
University of Technology

Table of contents

Title
Checkerboard copulas with maximum entropy: literature overview and a case study

1 Introduction

Dependence modeling between random variables is a widely studied subject in probability and statistics. In the context of statistics, by dependence, we mean a relationship between two or more random variables. When two random variables are dependent their values affect each other, i.e. if one random variable equals some value, the probability of the other dependent random variable becoming equal to some other value changes. An example where it is important to describe these dependencies is the model of the loss expectancy of a portfolio containing obligors. As there is dependence between the obligors defaulting their loan, banks and other institutions need to make accurate predictions of the amount of loss that could occur. One possible way to model dependencies in this and similar cases is by using copula functions[11]. In 1959 copulas were first mentioned in a paper by Sklar[12]. As copulas became more popular in the late 1990s, we can consider dependence modeling with copulas to be a modern subject within statistics. Copulas are useful tools to split the dependence structure and the marginal distributions of a random vector, and they can be used to construct flexible multivariate distribution functions. As stated, copulas can be used in several fields, including hydrology, medicine, and finance.
There are numerous practices for econometric estimation and ways to process data that are assumed to be multivariate normal distributed, but for joint nonlinear modeling of non-normal data, there is less knowledge[13]. This is where copulas come in handy, as they are well suited for joint nonlinear modeling of non-normal data. For the general theory of copula we refer the reader to Nelsen's book[6].

In this thesis, we look at how copulas, specifically a class of copulas named checkerboard copulas, can be used in the field of hydrology. In doing this we use the notion of a discrete copula.

As stated in the article of Kolesarova: discrete copulas are discrete bivariate distribution functions with uniform discrete marginals [3].

Discrete copulas are linked to a certain class of matrices called doubly-stochastic matrices, from such a matrix we can construct a discrete copula, this procedure can also be executed the other way around.

We can obtain a checkerboard copula by extending a discrete copula, using bilinear interpolation. Checkerboard copulas for multivariate modeling are discussed in 'Checkerboard copula defined by sums of random variables,' an article by Kuzmenko [14]. In this article, it is stated that a bivariate checkerboard copula is a distribution with corresponding density defined by a step function on a subdivision of the unit square. In our case study, we discuss how to find a checkerboard copula constrained to the correlation between the amount of total rainfall for two different months. This problem is similar to the optimization problem that is discussed in 'Copulas with maximum entropy' an article by Piantadosi et al. [9]. However, in their case study, they look at the rainfall for three different months. In our case study, we make use of the portfolio safeguard (PSG) software that is used in the article by Kuzmenko [14]. The problem that we discuss in this thesis is how to find a checkerboard copula with maximum disorder, or entropy, that matches a grade correlation coefficient, namely Spearman's rho coefficient. We calculate the correlation coefficient using historical data that we obtain from the KNMI [7], which is the Royal Dutch Meteorological Institute. The data is freely available, a link can be found in the bibliography. Models with a Checkerboard copula are mainly used to create synthetic data, to conduct further analysis related for example to risk assessment. Finding a checkerboard copula in our setting can be done by formulating and solving an optimization problem [9]. As we explain in the followin chapters, the reason we want to find the checkerboard copula is that it may form a compromise between dependence and independence, which makes it useful for modeling, as it is analytically simple but more accurate than just independence.

As stated in the article of Piantadosi, when monthly rainfall totals are treated as mutually independent random variables the standard deviation of simulated values becomes larger than that of the observed data [9]. Thus by implementing just the right amount of dependence, a rainfall model could be improved. Exactly this can be done by a checkerboard copula. In this thesis, we recall theory on checkerboard copulas and use it to model rainfall totals in Eindhoven for two different months. The reason we want a checkerboard copula instead of model fitting is to assess the quality of checkerboard copulas for simulating synthetic data in our case study context. Of course, if one has data modeling as final goal, other options might be preferable.

The goal of this thesis is to find a checkerboard copula for the total amount of rainfall in Eindhoven during two prescribed months, and to create a model with it.At last, we will also compare simulated values using our checkerboard as a model to the historical data. The structure of this thesis is as follows: In Chapter 2, we provide basic definitions on copulas. Then we present the most important theoretical result on copula theory, i.e. Sklar's theorem, which links copula functions with joint probability distributions. After that, we define discrete copulas. We show how a discrete copula can be constructed. At the end of chapter 2, we discuss measures of dependence and introduce Spearman's rho coefficient. In chapter 3 we define the bivariate checkerboard copula and show how we can compute the Spearman's rho coefficient of such a checkerboard copula. In chapter 4 we present our case study, and in chapter 5 we draw our conclusions.

2 Prerequisites on copula theory

2.1 Basic definitions

In this chapter we introduce copula functions, beginning with their formal definition[6].

Definition 2.1.1 *Let* $\boldsymbol{I} = [0,1]$*, then a function* $C : \boldsymbol{I} \times \boldsymbol{I} \to \boldsymbol{I}$ *a two-dimensional copula if the following properties hold:*

1. *For every* $u, v \in \boldsymbol{I}, C(u,0) = 0 = C(0,v)$
2. *For every* $u, v \in \boldsymbol{I}, C(u,1) = u$ *and* $C(1,v) = v$
3. *For every* $u_1, u_2, v_1, v_2 \in \boldsymbol{I}$*, such that* $u_1 \leq u_2$ *and* $v_1 \leq v_2$*,*

$$C(u_2, v_2) - C(u_2, v_1) - C(u_1, v_2) + C(u_1, v_1) \geq 0 \tag{2.1}$$

To visualize the third property of definition 2.1.1 we can take a look at the unit square in Figure 2.1.

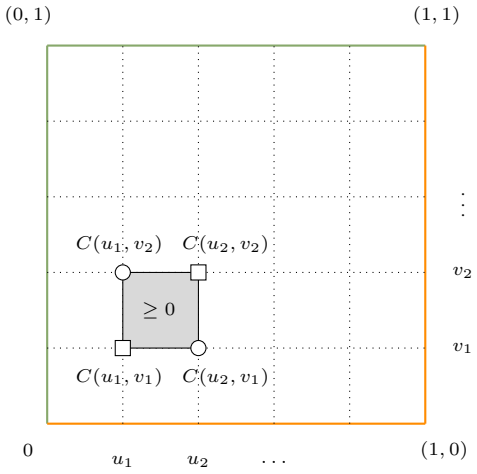


Figure 2.1: Visual representation of condition 2.1, in definition 2.1.1

Consider the grey square with four corners. The third property of Definition 2.1.1 simply means that the copula output of the upper right rectangle ($C(u_2, v_2)$) added with the bottom left rectangle ($C(u_1, v_1)$)is greater or equal than the copula output of the bottom right circle ($C(u_2, v_1)$) added with the upper left circle ($C(u_2, v_1)$). This is because the grey square ($C(u_2, v_2) - C(u_2, v_1) - C(u_1, v_2) + C(u_1, v_1)$) represents a probability, and therefore has to be larger than or equal to zero. Notice that for every arbitrary rectangle on the unit square, like the one in figure 2.1, the rectangle represents a probability measure and therefore is larger or equal than zero. Now, to clarify what a copula might look like, we show examples of simple copula functions.

Example 2.1.1 *The function* $M(u,v) = min(u,v)$ *is a copula. Indeed we can show that the properties of Definition 2.1.1 hold.*

$min(0,v) = 0 = min((u,0)$ *thus the first property of Definition 2.1.1 holds.*

$min(u,1) = u$ *and* $min(1,v) = v$*, note that* $u, v \in [0,1]$ *thus the second property of Definition 2.1.1 holds.*

Take $u_1, u_2, v_1, v_2 \in \boldsymbol{I}$*, such that* $u_1 \leq u_2$ *and* $v_1 \leq v_2$*, then* $min(u_2, v_2) - min(u_2, v_1) - min(u_1, v_2) + min(u_1, v_1)$ *has four possible outcomes:*

1. *If* $u_2 \geq v_2 \geq u_1 \geq v_1$*, we obtain* $v_2 - v_1 - u_1 + v_1 = v_2 - u_1 \geq 0$
2. *If* $u_2 \geq u_1 \geq v_2 \geq v_1$*, we obtain* $v_2 - v_1 - v_2 + v_1 = 0$
3. *If* $v_2 \geq u_2 \geq v_1 \geq u_1$*, we obtain* $u_2 - v_1 - u_1 + u_1 = u_2 - v_1 \geq 0$
4. *If* $v_2 \geq v_1 \geq u_2 \geq u_1$*, we obtain* $u_2 - u_2 - u_1 + u_1 = 0$

Here it is important to notice the symmetry. Using these four outcomes we are able to conclude that the third property of Definition 2.1.1 is satisfied as well, thus $M(u,v) = min(u,v)$ *is a copula.*
The functions $W(u,v) = max(u+v-1,0)$ *and* $\Pi(u,v) = uv$ *are also copulas. This can be proven in the same way as for* $M(u,v) = min(u,v)$.

In Figure 2.2 we see a visual representation of the three copulas we have discussed. Figure 2.2 is from Nelsen's book [6].

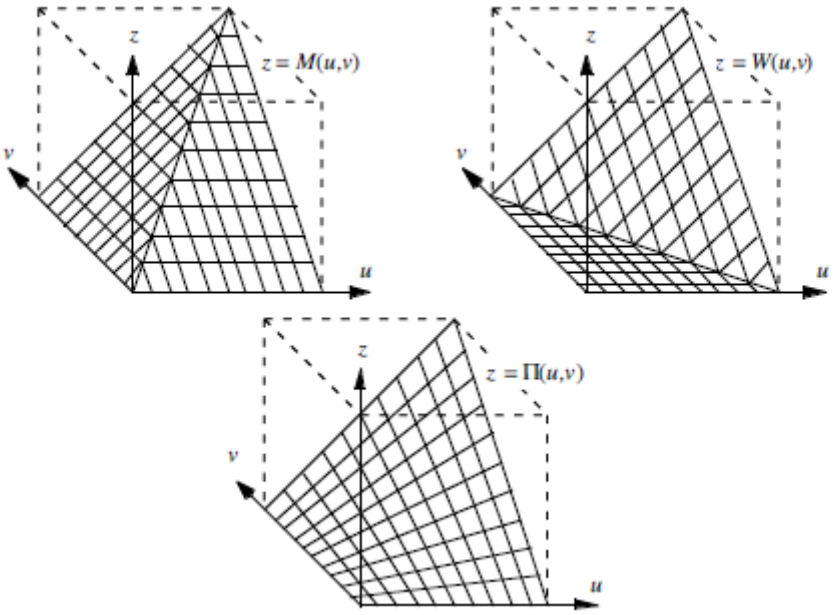


Figure 2.2: A picture of the copula functions $M(u,v), W(u,v)$ and $\Pi(u,v)$, by Nelsen [6].

The two copula functions $W(u,v) = max(u+v-1,0)$ and $M(u,v) = min(u,v)$ are special copulas, they are respectively called the Fréchet-Hoeffding upper and lower bound. This is because for every copula we have the following inequality.

$$W(u,v) \leq C(u,v) \leq M(u,v) \tag{2.2}$$

For the proof of this inequality we refer readers to Theorem 2.2.3 of Nelsen's book[6].

2.2 Sklar's theorem

In this section, we present Sklar's theorem, which is the most important result on copulas as it provides the theoretical foundation for their use in applications. For more details, we refer the reader to Nelsen's book.[6]

Theorem 2.2.1 (Sklar's theorem) *Let H be a joint distribution function with margins F and G. Then there exists a copula C such that for all* $x, y \in \mathbb{R}$

$$H(x, y) = C(F(x), G(y))$$

If F and G are continuous, then C is unique; otherwise, C is uniquely determined on $ranF \times ranG$*. Conversely, if we have a copula C, and F and G distribution functions, then we have that* $H(x, y) = C(F(x), G(y))$ *is a joint distribution function with margins* F *and* G*.*

Note that with $ranF$ and $ranG$ we mean the range of F and G, which are both $[0, 1]$ if F and G are continuous. If F and G are not continuous as a result of Sklar's theorem we obtain uniqueness of the copula associated with H on a subset of the unit square. More specifically if F and G are both discrete, by Sklar's theorem we obtain a copula that is uniquely defined on a grid. Copulas that are defined on a grid are called discrete copulas. In the next section, we will discuss these discrete copulas and show a general definition for them.

2.3 Discrete copulas

We start the section with the general definition for discrete copulas as it is introduced in Discrete Copulas, an article by A. Kolesárová, R. Mesiar, J. Mordelová, and C. Sempi [3].

Definition 2.3.1 *Given* $\boldsymbol{I}_n = \{0, 1/n, ..., n/n\}$ *and* $\boldsymbol{I}_m = \{0, 1/m, ..., m/m\}$ *with* $n, m \in \mathbb{N}$*, then* $C_n : \boldsymbol{I}_n \times \boldsymbol{I}_m \to [0, 1]$ *is a discrete copula if and only if the following properties hold for all* $i \in 0, ..., n$ *and* $j \in 0, ..., m$

1. $C_{n,m}(i/n, 0) = C_{n,m}(0, j/m) = 0$
2. $C_{n,m}(i/n, 1) = i/n, C_{n,m}(1, j/m) = j/m$
3. *While for all* $i \in 1, ..., n$ *and* $j \in 1, ..., m$

$$C_{n,m}(i/n, j/m) + C_{n,m}((i-1)/n, (j-1)/m) - C_{n,m}((i-1)/n, j/m) - C_{n,m}(i/n, (j-1)/m) \geq 0 \quad (2.3)$$

We notice that the definition is very similar to the general definition for a copula, but for a different domain. A useful property of square discrete copulas is that we can obtain them from a special type of matrices, namely, doubly-stochastic matrices, defined as follows. A square discrete copula is a copula with $n = m$, thus the sets I_n is identical to the set I_m.

Definition 2.3.2 *A doubly-stochastic matrix is a matrix such that all rows and columns add up to one, i.e. let* $H = (h_{ij}$ *be a* $(n \times n)$ *matrix, then* $\sum_{i=1}^{n} h_{ij} = \sum_{k=1}^{n} h_{jk} = 1$ *for all* $j = 1, ..., n$*.*

Proposition 1 *[3] For a function* $C_n : (I_n) \to [0, 1]$ *the following statements are equivalent:*

1. C_n *is a discrete copula*
2. *there is a doubly-stochastic matrix* $A = (a_{ij})_{i,j=1}^{n}$ *such that for* $i, j \in 0, 1, 2, ..., n$

$$(c_{i,j})^{(n)} := C_n(i/n, j/n) = 1/n \sum_{k=1}^{i} \sum_{m=1}^{j} a_{km} \quad (2.4)$$

Notice that by using the procedure from Proposition 1 we would obtain a discrete copula $C_{n,m}$ with $n = m$, therefore we write this copula simply as C_n. Conversely we can also derive a doubly-stochastic matrix $A = (a_{ij})_{i,j=1}^{n}$ from a square discrete copula.

$$a_{ij} = n((c_{i,j})^{(n)} - (c_{i-1,j})^{(n)} - (c_{i,j-1})^{(n)} + (c_{i-1,j-1})^{(n)}) \quad (2.5)$$

, with $i, j \in 0, 1, 2, ..., n$

Notice that equation 2.5 looks very similar to equation 2.3, using equation 2.3 we can conclude that $a_{ij} \geq 0$.

Equation 2.4 is very useful to construct discrete copulas[3]. We will provide two examples to show how this procedure and the procedure from equation 2.5 can be used.

Example 2.3.1 *Let B be a doubly-stochastic matrix defined as:*

$$B = \begin{pmatrix} 1/5 & 2/5 & 2/5 \\ 3/5 & 1/5 & 1/5 \\ 1/5 & 2/5 & 2/5 \end{pmatrix}$$

By applying the transformation of Proposition 1 we obtain $c_{11} = 1/3 \cdot 1/5 = 1/15$, $c_{12} = 1/3 \cdot (1/5 + 2/5)$, $c_{22} = 1/3 \cdot (1/5 + 2/5 + 3/5 + 1/5) = 7/15$, etc.
Therefore the discrete copula associated with B is

$$C = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1/15 & 1/5 & 1/3 \\ 0 & 4/15 & 7/15 & 2/3 \\ 0 & 1/3 & 2/3 & 1 \end{pmatrix}$$

Note that with this matrix we mean that $C(0.25, 0.25) = 1/15$, and $C(0.5, 0.25) = 4/15$ etc.

The other way around we can create a doubly-stochastic matrix from a discrete copula.

Example 2.3.2 *Let C_1 be a discrete copula defined as:*

$$C_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1/18 & 1/9 & 1/3 \\ 0 & 1/6 & 4/9 & 2/3 \\ 0 & 1/3 & 2/3 & 1 \end{pmatrix},$$

then by applying equation 2.5 we obtain

$$B_1 = \begin{pmatrix} 1/6 & 1/6 & 2/3 \\ 1/3 & 2/3 & 0 \\ 1/2 & 1/6 & 1/3 \end{pmatrix}$$

A special type of discrete copulas, called empirical discrete copulas, can be directly constructed from a data set.

An empirical discrete copula C_n is defined as follows

$$C_n(i/n, j/n) = \frac{(\#(x, y) | x \leq x_{(i)} \text{ and } y \leq y_{(j)})}{n} \tag{2.6}$$

here $x_{(i)}$ and $y_{(j)}$, with $1 \leq i, j \leq n$ represent the rank of the sample, with the sample we mean an entry from the data set. $\#$ represents the cardinality, or the number of elements, of the set. So in words, the empirical discrete copula value, with $i, j \in (1, ..., n)$, equals the number data entries for which we have that both the x value and the y value is smaller than or equal to the value of respectively $x_{(i)}$ and $y_{(j)}$.

We consider the small data set in Table 2.1, where we have three different persons with their weight and height.

person	height (cm)	weight (kg)
1	183	66
2	182	68
3	167	57

Table 2.1: data set

We now follow the steps presented in [5] to construct the corresponding empirical copula.

Example 2.3.3 *Let us define the weight as random variable X, and the height as random variable Y. Then we determine the rank of the different values in the table, so we obtain table 2.2 including ranks, these ranks are defined as the order of our data values. So the smallest value corresponds to rank one, and the largest to rank three.*

Then we determine the values for our empirical copula using the following procedure: $C_{ij} = \boldsymbol{P}(X <$ (value for X with rank i)$, Y <$ (value for Y with rank j)$)$ So for example we obtain $c_{22} = \boldsymbol{P}(X < 182, Y < 66) = 1/3$, and $c_{34} = \boldsymbol{P}(X < 183, Y \leq 68) = 2/3$. Also note that for the first row and column we obtain only zero values. Using the data set and the procedure that is described we obtain the following discrete copula:

person	X height (cm)	Y weight (kg)	rank X	rank Y
1	183	66	3	2
2	182	68	2	3
3	167	57	1	1

Table 2.2: data set including ranks

$$C_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1/3 & 1/3 & 1/3 \\ 0 & 1/3 & 1/3 & 2/3 \\ 0 & 1/3 & 2/3 & 1 \end{pmatrix}$$

In order to obtain the corresponding doubly-stochastic matrix it is more convenient to use the version that does include the first column and row. Then we may obtain the following doubly-stochastic matrix by using the procedure described in Equation 2.6.

$$B_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Note that this doubly-stochastic matrix contains only ones and zeros, which is called a permutation matrix. It is a property of empirical copula that their corresponding doubly-stochastic matrix is a permutation matrix.

At last, we would like to mention that there are infinite ways of extending a discrete copula to a full domain copula, i.e. such that it is defined on the whole unit square $[0, 1]^2$. One of these possible extensions will transform a discrete copula into a checkerboard copula. This extension to a checkerboard copula is bilinear interpolation. The checkerboard copula is defined on all the squares that make up the grid, also all these partitions are connected. As stated previously copulas are a tool to describe the dependence between random variables. We would like to have some way to define the amount of dependence a data set or a copula function has. Using this we can create a copula with the right amount of dependence. For the data set in Table 2.1 and Table 2.2, we are also able to define the amount of dependence it contains. This will be discussed in the next section.

2.4 Measures of Dependence

The level of dependence between two random variables can be measured in several ways, one of which is by the computation of Spearman's rho. In order to define Spearman's rho we need to introduce the notion of concordance. We define concordance as follows.

Definition 2.4.1 *[6] Take two continuous random variables X and Y, with observations (x_i, y_i) and (x_j, y_j). Then (x_i, y_i) and (x_j, y_j) are concordant if $x_i < x_j$ and $y_i < y_j$, or (x_i, y_i) and (x_j, y_j) are discordant if $x_i > x_j$ and $y_i > y_j$. Alternatively we may also say that (x_i, y_i) and (x_j, y_j) are concordant if $(x_i - x_j)(y_i - y_j) > 0$ and discordant if $(x_i - x_j)(y_i - y_j) < 0$*

In words, two random variables are concordant if large values of one are associated with large values of the other, and small values of one with small values of the other.

Then we can define a function that gives the difference between the probabilities of concordance and discordance of our observations (x_i, y_i) and (x_j, y_j).

Definition 2.4.2 *Let (X_i, Y_i) and (X_j, Y_j) be independent vectors of continuous random variables with joint distribution functions H_i and H_j, respectively, with common margins F of X_i and X_j and G of Y_i and Y_j. Let C_i and C_j denote the copulas of (X_i, Y_i) and (X_j, Y_j), respectively, such that $H_i(x, y) = C_i(F(x), G(y))$ and $H_2(x, y) = C_2(F(x), G(y))$. Note that this is a result from Sklar's theorem 2.2.1. Then we may define function Q which denotes the difference between the probability of concordance and discordance of (X_i, Y_i) and (X_j, Y_j) as follows.*

$$Q = P[(X_i - X_j)(Y_i - Y_j) > 0] - P[(X_i - X_j)(Y_i - Y_j) < 0]$$

From this we may conclude that

$$Q = Q(C_1, C_2) = 4 \int \int_{\mathbf{I}^2} C_2(u, v) dC_1(u, v) - 1$$

person	X height (cm)	Y weight (kg)	rank X	rank Y	d
1	183	66	3	2	1
2	182	68	2	3	1
3	167	57	1	1	0

Table 2.3: data set including difference between ranks

For a more detailed explanation we would like to refer readers to Nelsen's book [6]. To obtain the population version of Spearman's rho coefficient we let (X_1, Y_1), (X_2, Y_2) and (X_3, Y_3) be three independent random vectors with common joint distribution function H[6]. Then the population version of Spearman's rho coefficient is proportional to difference of the probability of concordance and the probability of discordance for the random vectors (X_1, Y_1) and (X_2, Y_3). Thus we obtain that Spearman's rho coefficient is defined as follows:

$$\rho_{X,Y} = 3(P[(X_1 - X_2)(Y_1, Y_3) > 0] - P[(X_1 - X_2)(Y_1 - Y_3) < 0])$$

Note that as X_2 and Y_3 are independent so their corresponding copula is the product copula Π. Then using the result of definition 2.4.2 we have

$$\begin{aligned} \rho_{X,Y} &= \rho_C = 3Q(C, \Pi), \\ &= 12 \int \int_{\mathbf{I}^2} uvdC(u, v) - 3, \\ &= 12 \int \int_{\mathbf{I}^2} C(u, v)dudv - 3 \end{aligned}$$

Notice that this expression can also be rewritten as

$$12(E[UV] - 1/4) \tag{2.7}$$

Here we define U and V respectively as $F(X)$ and $G(Y)$, which are uniform random variables on $[0, 1]$ and have joint distribution function C. $F(X)$ and $G(Y)$ are uniform because they represent the grades of respectively x and y[6]. With grade we mean the population version of ranks.

Example 2.4.1 *As stated in the previous section we are able to define a Spearman's rho value for the data set represented by table 2.2. This we call an empirical Spearman's rho, which is associated with the empirical discrete copula. To make this more clear we add one more column, for the difference between the ranks for each person. Therefore we obtain table 2.3.*

Then to calculate the spearman's rho value we use the following expression[6]:

$$\rho = 1 - 6\frac{\sum d_i^2}{n(n^2 - 1)}$$

From this we are able to obtain that $\rho = \frac{1}{2}$.

Notice that if X and Y are continuous random variables, and we have that $E[U] = E[V] = 1/2$, and $Var(U) = Var(V) = 1/12$, then their Spearman's rho value will be equal to Pearson's correlation coefficient.

$$12(E[UV] - 1/4) = \frac{E[UV] - 1/4}{1/12} = \frac{E[UV] - E[U]E[V]}{\sqrt{Var(U)}\sqrt{Var(V}}$$

Pearson's correlation coefficient of U and V is defined as

$$\frac{cov(UV)}{\sigma_U \sigma_V} \tag{2.8}$$

Which can be rewritten as

$$\frac{E[UV] - E[U]E[V]}{\sqrt{Var(U)}\sqrt{Var(V)}}$$

Thus we have shown that the Pearson's correlation coefficient equals the Spearman's rho value if X and Y are both uniformly distributed on $[0, 1]$.

Now as we have become more familiar with copulas, discrete copulas, and dependence, we may introduce the checkerboard copula. After defining a checkerboard copula we will derive the Spearman's rho coefficient that corresponds with it.

3 Bivariate checkerboard copulas

A bivariate checkerboard copula is a distribution with a density defined almost everywhere by a step function on a subdivision of the unit square. As mentioned in the introduction a step function is a function that describes a radical change. Thus the density of a checkerboard copula is constant inside a part of the grid, and might be different in the other parts. The radical change is between those two parts, the density does not for example linearly increases or decreases, but it instantly changes to the other value.
A practical way to define a checkerboard copula is by using a $n \times n$ matrix similar to a doubly-stochastic matrix. However, for this special type of doubly-stochastic matrix, the sum of all entries equals n, and the sum of all rows and columns equals one.
An example of this special type of doubly-stochastic matrix is matrix B_3.

$$B_3 = \begin{pmatrix} 1/6 & 2/3 & 1/6 \\ 1/3 & 1/3 & 1/3 \\ 1/2 & 0 & 1/2 \end{pmatrix}$$

Then we are able to compute the copula values using proposition 1 in section 2.3.

As the rows and columns add up to one, we can also normalize B_3 to obtain a density distribution. So within all of the nine tiles in which we have divided the unit square, we are aware of the total distribution, however, within all of these tiles, the density can be distributed in infinitely many ways. Here checkerboard copulas are simply the case in which the density within all of the different squares is constant, and the probability mass function is distributed uniformly.
Now that we have some understanding of what a checkerboard copula is we will show a more formal mathematical definition. In the remainder of this section, we follow Kuzmenko's paper [14], but we only consider the two-dimensional case.

Let (X_1, X_2) be a pair of real-valued random variables and let $g(x_1, x_2)$ be the joint probability density, then we obtain the following marginal probability densities:

$$g_1(x_1) = \int_{\mathbb{R}} g(x_1, x_2)\, dx_2$$
$$g_2(x_2) = \int_{\mathbb{R}} g(x_1, x_2)\, dx_1$$

for $x_1 \in \mathbb{R}$ and $x_2 \in \mathbb{R}$

Then let $c(u_1, u_2)$ be a density of a two-dimensional copula, so

$$c_1(u_1) = \int_0^1 c(u_1, u_2)\, du_2 = 1$$
$$c_2(u_2) = \int_0^1 c(u_1, u_2)\, du_1 = 1$$

for all $u_1 \in [0, 1]$ and all $u_2 \in [0, 1]$
Let $g_1(x_1)$ and $g_2(x_2)$ be the known probability densities with corresponding cumulative distribution functions $F_1(x_1)$ and $F_2(x_2)$ for real-valued random variables X_1 and X_2. The joint density, defined by the copula density $c(u_1, u_2)$, is defined as follows, for $(x_1, x_2) \in \mathbb{R}^2$,

$$g(x_1, x_2) = c(F_1(x_1), F_2(x_2))g_1(x_1)g_2(x_2) \tag{3.1}$$

Note that equation 3.1 results from Sklar's Theorem, Theorem 2.2.1 in this thesis.

Now we take a doubly-stochastic matrix $H = (h_{ij})$, and we define a partition $a(k) = (k-1)/n$ of the interval $[0, 1]$,with $k = 1, ..., n+1$ such that $0 = a(1) < a(2) < ... < a(n+1) = 1$.
Define the step function $c(u_1, u_2)$ almost everywhere on the region $[0, 1] \times [0, 1]$ by $c(u_1, u_2) = nh_{ij}$ for $(u_1, u_2) \in (a(i), a(i+1)) \times (a(j), a(j+1)), i = 1, ..., n, j = 1, ..., n$

This last line simply represents a rectangle with $u_1 \in [a(i), a(i+1)]$ and $u_2 \in [a(j), a(j+1)]$.

By integrating $c(u_1, u_2)$ over the unit plane equals one, we conclude that $c(u_1, u_2)$ is a joint density function. Indeed, the following holds.

$$\int_0^1 \int_0^1 c(u_1, u_2)\, du_1 du_2 = \sum_{i=1}^{n} \sum_{i=1}^{n} nh_{ij} \int_{a(i)}^{a(i+1)} \int_{a(j)}^{a(j+1} du_1 du_2 =$$

$$\sum_{i=1}^{n}\sum_{i=1}^{n} nh_{ij}1/n^2 = 1/n\sum_{i=1}^{n}\sum_{i=1}^{n} h_{ij} = 1/n\sum_{i=1}^{n} 1 = 1$$

Suppose $u_1 \in [a(i), a(i+1)]$, then

$$\int_0^1 c(u_1,u_2)du_2 = \sum_{j-1}^{n}\int_{a(j)}^{a(j+1)} nh_{ij}du_2 = \sum_{j=1}^{n} nh_{ij}\int_{a(j)}^{a(j+1)} du_2 = \sum_{j=1}^{n} nh_{ij}1/n = \sum_{j=1}^{n} h_{ij} = 1$$

Therefore the marginal densities, for $u_1 \in [0,1]$ and $u_2 \in [0,1]$, equal

$$c_1(u_1) = \int_0^1 c(u_1,u_2)du_2 = 1$$
$$c_2(u_2) = \int_0^1 c(u_1,u_2)du_1 = 1$$

Then, as we have shown we can define a density function of a copula by using a doubly-stochastic matrix, we are able to define the checkerboard copula by integration.

We obtain the following checkerboard copula, $C : [0,1]\times[0,1] \to [0,1]$

$$C(u_1,u_2) = \int_0^{u_1}\int_0^{u_2} c(v_1,v_2)dv_1dv_2$$

This construction we can use for the density, but to construct an expression for a checkerboard copula we can also extend a discrete copula to a checkerboard copula using the following Lemma.

Lemma 1 *Let DC be a discrete copula. Then there exists a checkerboard copula CC such that $CC(u,v) = DC(u,v)$ for all (u,v) in the domain of DC.*

This checkerboard copula can be constructed using the following procedure. Let $DomDC = I_n^2$. Now we extend DC to a function CC with domain $[0,1]^2$. Let (a,b) be any point in $[0,1]^2$, let a_1 and a_2 be, respectively, the greatest and least elements of S_1 that satisfy $a_1 \le a \le a_2$; and let b_1 and b_2 be, respectively, the greatest and least element of S_2 that satisfy $b_1 \le b \le b_2$. Note that if a is in S_1, then $a_1 = a = a_2$; and if b is in S_2, then $b_1 = b = b_2$. Now let

$$\lambda_1 = \begin{cases} (a-a_1)/(a_2-a_1), & \text{if } a_1 < a_2 \\ 1, & \text{if } a_1 = a_2 \end{cases}$$

$$\mu_1 = \begin{cases} (b-b_1)/(b_2-b_1), & \text{if } b_1 < b_2 \\ 1, & \text{if } b_1 = b_2; \end{cases}$$

and define

$$\begin{aligned} CC(a,b) = (1-\lambda_1)(1-\mu_1)DC(a_1,b_1) + (1-\lambda_1)\mu_1 DC(a_1,b_2) \\ + \lambda_1(1-\mu_1)DC(a_2,b_1) + \lambda_1\mu_1 DC(a_2,b_2) \end{aligned} \tag{3.2}$$

Lemma 1 is similar to Lemma 2.3.5 from Nelsen's book[6], except here we have a discrete copula and a checkerboard copula instead of a subcopula and a copula. The procedure of Equation 3.2 is a bilinear interpolation.

3.1 Spearman's rho coefficient of a checkerboard copula

As we want to find a checkerboard copula for a fixed Spearman's rho value, we should define the Spearman's rho value for checkerboard copulas, we will do this in this section. We recall the last expression we obtained for Spearman's rho coefficient of a regular copula in equation

$$12(E[UV] - 1/4)$$

with U and V respectively as $F(X)$ and $G(Y)$. We now calculate it from the density matrix of a checkerboard copula. Since $c_h(x,y) = n \cdot h_{xy}$, we obtain that

$$E[UV] = \int_0^1\int_0^1 c(u,v)\cdot uv \cdot dudv$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} \int_{u_i}^{u_{i+1}} \int_{v_j}^{v_{j+1}} nh_{ij} \cdot uv \cdot dudv$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} nh_{ij}(\int_{u_i}^{u_{i+1}} u \cdot du)(\int_{v_i}^{v_{i+1}} v \cdot dv)$$

$$= 1/n^3 \cdot \sum_{i=1}^{n}\sum_{j=1}^{n} h_{ij}(i-1/2)(j-1/2)$$

Therefore the coefficient correlation of a checkerboard copula is given by

$$\rho = 12(1/n^3 \cdot \sum_{i=1}^{n}\sum_{j=1}^{n} h_{ij}(i-1/2)(j-1/2) - 1/4)$$

Here we made use of the definitions given in Piantadosi et al. [8].

3.2 Entropy of a bivariate checkerboard copula

Entropy is used to describe the amount of disorder or randomness. We show the general formula for entropy, after which we will define the entropy of a bivariate checkerboard copula.

Let X be a discrete random variable, with possible outcomes $x_1, ..., x_n$, with probability $\mathbb{P}(x_1), ..., \mathbb{P}(x_n)$ of occurring. Then we define the entropy of X as follows.

$$H(X) = -\sum_{i=1}^{n} \mathbb{P}(x_i) log(\mathbb{P}(x_i) \tag{3.3}$$

For a bivariate checkerboard copula may define entropy as follows[8]. Let $h \in \mathbb{R}^2$ be a doubly-stochastic matrix and let $c_h : [0,1]^2 \to \mathbb{R}$ be the associated elementary joint density. The entropy of h than is defined as follows:

$$J(h) = (-1) \int_{[0,1]^2} c_h(u) log c_h(u) \cdot du \tag{3.4}$$

$$= (-1) \sum_{i \in \{1,...,n\}^2} (nh_i) \cdot log(nh_i) \cdot 1/n^2$$

$$= (-1)1/n \sum_{i \in \{1,...,n\}^2} h_i(logh_i + logn)$$

$$= -1(1/n \sum_{i \in \{1,...,n\}^2} h_i log(h_i) + log(n))$$

For a system with a finite number of possible states, like a discrete random variable, the entropy is maximized when all probabilities are equal [1]. Thus if the probability of an event occurring equals 1 divided by the amount of events, for all possible events, we have maximized entropy. In our case the entropy would be maximized if all entries of matrix $H = (h_{ij})$ are equal, their value would be 1 divided by the amount of entries. In the case study we maximize entropy with a prescribed Spearman's rho coefficient, and by doing so we find a checkerboard copula[14]. Here the entropy of matrix h is defined by:

$$J(h) = -1(1/n \sum_{i \in \{1,...,n\}^2} h_i log(h_i) + log(n)) \tag{3.5}$$

As we have a prescribed Spearman's rho coefficient, that is a constraint of our optimization problem, we will not obtain a matrix for which all entries are equal. The prescribed Spearman's rho coefficient will keep some dependence in the checkerboard copula, and the maximization of the entropy will make it as independent as possible considering the Spearman's rho coefficient. Therefore we will obtain a checkerboard copula that is a compromise between dependence and independence.

4 Case study

4.1 Data analysis

We now consider data of the total amount of rainfall for all the months in Eindhoven. We look at data from 1951 until 2010. Summary statistics of this data can be found in table 4.1. We notice that in April the amount of rain is significantly lower than that of most other months.

Type	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Min	4.9	2.9	4.5	0.2	3.5	15.9	12.4	11.0	4.3	4.1	8.8	14.1
1st Qu	43.9	28.4	39.0	26.7	39.57	45.2	40.7	43.8	35.4	39.2	45.3	48.7
Median	66.7	54.6	53.7	44.5	60.2	62.2	71.3	65.3	53.5	62.9	72.5	70.5
Mean	68.0	55.1	60.7	46.8	61.4	65.1	76.7	71.5	61.6	64.2	72.7	73.4
3rd Qu	91.2	74.2	83.1	66.6	80.0	80.0	106.2	92.9	75.0	85.8	92.2	92.1
Max.	157.9	128.4	141.5	100.9	135.3	148.2	197.1	180.8	164.5	174.0	166.8	159.0

Table 4.1: Summary statistics of our data, the total monthly amount of rainfalln(in mm) in Eindhoven van 1951 until 2010

We noticed that January and September show the most correlation, therefore we will consider these months. The scatterplots of rainfall of these months is shown in Figure 4.1. The Spearman's rho coefficient of January and September is 0.426. What is also remarkable, is that in July, even though the mean is not the largest value compared to the other months, the third quarter value and the maximum value are significantly larger than those of the other months. What we may conclude from this is that in July it is more likely to have an extremely large amount of rain.

4.2 Problem statement

Now that we have obtained Spearman's rho coefficient we have a constraint for the optimization problem, using this we need to find our matrix with maximum entropy. Our optimization problem is defined as follows:

Problem 1 *Find matrix* $\boldsymbol{h} \in \mathbb{R}^2$ *by maximizing*

$$J(\boldsymbol{h}) = -1(1/n \sum_{i \in \{1,\dots,n\}^2} h_i log(h_i) + log(n)) \tag{4.1}$$

subject to the following constraints:

$$\rho = 12/n^3 \cdot (\sum_{i \in \{1,\dots,n\}^2} h_{i,i}(i - 1/2)^2) - 3 \tag{4.2}$$

$$\sum_{\boldsymbol{i} \in \{1,\dots,n\}^2} h_{\boldsymbol{i}} = 1 \tag{4.3}$$

with $i \in \{1, ..., n\}$

$$h_{\boldsymbol{i}} \geq 0 \tag{4.4}$$

with $\boldsymbol{i} \in \{1, ..., n\}^2$

To solve this problem we use the Portfolio safeguard (PSG) software. This is a tool that can be used for optimization problems. PSG can be used in three different programming languages, Matlab[4], Run-file, and R. We use the software in the language R[10]. As the name suggests PSG is mostly used for finance and risk management problems [2]. However, it is also well suited to be used for our hydrology case study. The software is open source for academic use. Our goal is to find a checkerboard copula with maximum entropy and the same Spearman's rho coefficient as our historical data.

4.2.1 PSG Software

To solve our optimization problem we make use of the PSG software in R [10]. In the paper from Kuzmenko [14] for the constraint matrix of the data they consider a five-dimensional hyper-matrix with a partition size of four. In

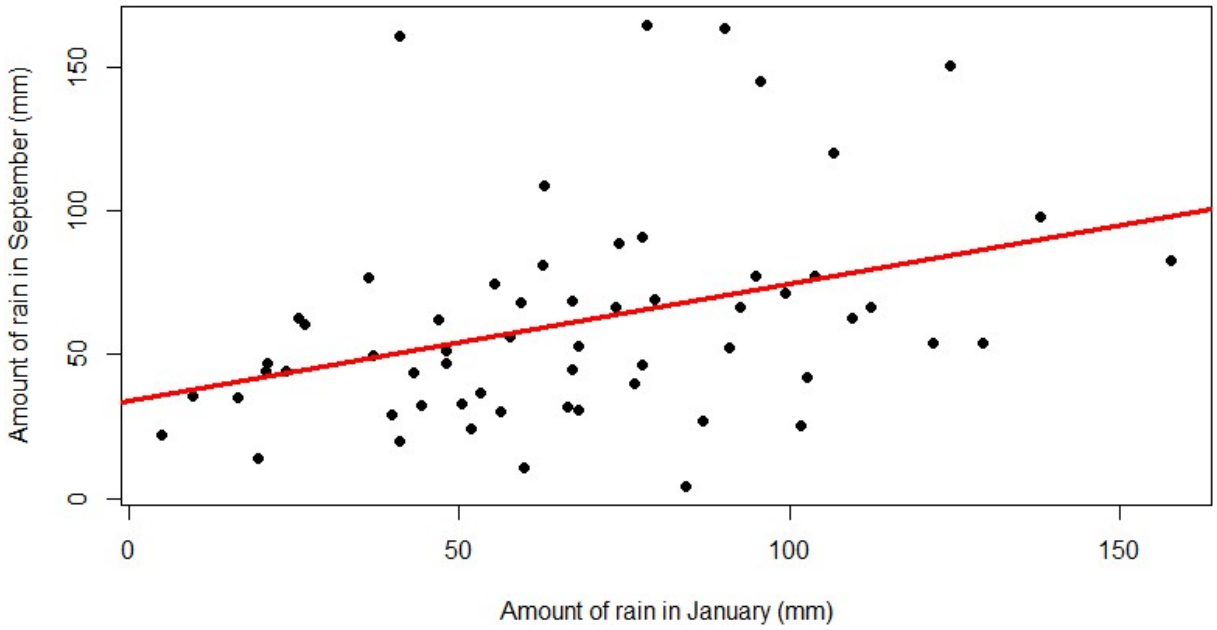


Figure 4.1: Correlation between the rainfall of January and September

our case study, we simplify this by considering only two random variables, rainfall in January and September. As in Kuzmenko [14], we assume n is equal to four. The values of the constraint matrix are based upon the formula for the Spearman's rho value:

$$\rho = 12/n^3 \cdot (\sum_{i \in \{1,\dots,n\}^2} h_i(i - 1/2)^2) - 3$$

More precisely the entries of the constraint matrix are as follows:

$$12/n^3(i - 1/2)^2$$

and the benchmark value equals

$$-(\rho + 3)$$

which in our case equals -3.426. The benchmark value ensures that the checkerboard copula we find will be constrained to the Spearman's rho value of 0.426.

4.3 Simulating data using the checkerboard copula

The result we obtained by executing the software is the following doubly-stochastic matrix B_4, which is represented as a 4×4 matrix, as we are dealing with a bivariate problem. Note that this matrix is rounded to four decimals, in the calculations we use sixteen decimals.

$$B_4 \approx \begin{pmatrix} 0.4684 & 0.3026 & 0.1585 & 0.0705 \\ 0.2987 & 0.2936 & 0.2492 & 0.1585 \\ 0.1618 & 0.2420 & 0.2936 & 0.3026 \\ 0.0711 & 0.1618 & 0.2987 & 0.4684 \end{pmatrix}$$

This result is our starting point to make simulated data. Then, we compare the simulated data to the historical data we have analyzed. We proceed by constructing the discrete copula corresponding to the doubly-stochastic matrix B_4. This is done using Proposition 1 from the discrete copula section, and it results in:

$$C_4 \approx \begin{pmatrix} 0.1171 & 0.1918 & 0.2323 & 0.25 \\ 0.1927 & 0.3408 & 0.4418 & 0.5 \\ 0.2324 & 0.4427 & 0.6171 & 0.75 \\ 0.25 & 0.5 & 0.75 & 1 \end{pmatrix}$$

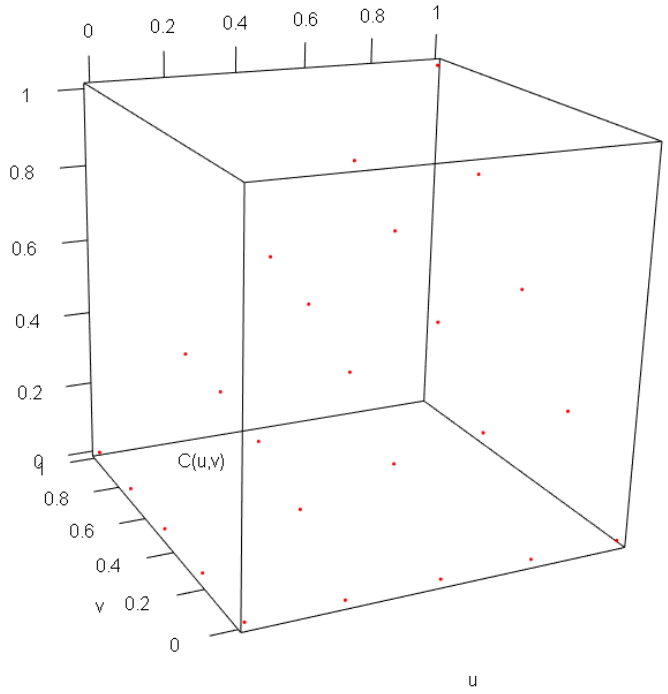


Figure 4.2: A three-dimensional figure of the discrete copula

Example 4.3.1 *In this example we will show how two of the sixteen expressions for each square grid partition of the checkerboard copula, associated with C_4, were constructed.*
For the first expression we consider the plane for $0 \leq a \leq 0.25$ and $0 \leq b \leq 0.25$,
as $C_4(0,b) = C_4(a,0) = 0$, $\lambda_1 = 4a$ and $\mu_1 = 4b$,
we obtain

$$CC(a,b) = 16abC_4(0.25, 0.25)$$

. Since $C_4(0.25, 0.25) \approx 0.1171$, we have that

$$CC(a,b) \approx 1.8737ab$$

Thus

$$\frac{\partial CC}{\partial a} := c_a(b) = 1.8737b$$

For the second expression we consider the plane for $0.5 < a \leq 0.75$ and $0.25 < b \leq 0.5$,
we have $\lambda_1 = 4a - 2$ and $\mu_1 = 4b - 1$.
Using this we obtain

$$\begin{aligned} C(a,b) = {} & C_4(0.5, 0.25)(16ab - 8a - 12b + 6) + C_4(0.5, 0.5)(-16ab + 4a + 12b - 3) \\ & + C_4(0.75, 0.25)(-16ab + 8a + 8b - 4) + C_4(0.75, 0.5)(16ab - 4a - 8b + 2) \end{aligned}$$

and

$$\begin{aligned} \frac{\partial CC}{\partial a} := c_a(b) = {} & 16b(C_4(0.5, 0.25) - C_4(0.5, 0.5) - C_4(0.75, 0.25) + C_4(0.75, 0.5)) \\ & + 4(-2C_4(0.5, 0.25) + C_4(0.5, 0.5) + 2C_4(0.75, 0.25) - C_4(0.75, 0.5)) \end{aligned}$$

Notice that at the boundaries the checkerboard copula has the same values as the discrete copula, so for example $CC(0.5, 0.5) = C_4(0.5, 0.5) \approx 0.3408$.

As discussed in Chapter 3, we can extend the discrete copula C_4 to a full domain checkerboard copula. This is done by constructing planes between the points at which the discrete copula is defined. This process is bilinear interpolation, and is described in Lemma 1. The result of this procedure can be seen in Figure 4.3. Note that in the figure the planes are not filled with color, however, the checkerboard copula is defined in all the planes between the red lines shown. What we mean with this is that in Figure 4.3 we only see the edges of each plane, which are red lines, and the planes themselves are transparent. However, the checkerboard copula is defined in the whole of all planes. Now that we have obtained our checkerboard copula we are able to sample values, on the

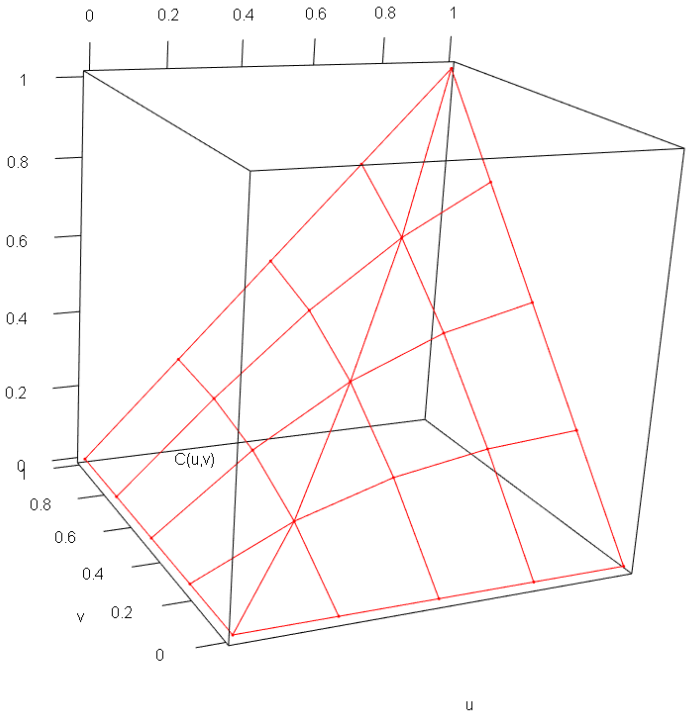


Figure 4.3: A three-dimensional figure of the checkerboard copula

unit square that represent the rainfall for January and September, using the algorithm described in section 2.9 of Nelsen's book[6]. We sample two uniform values: u and t. Then we calculate the value of v such that

$$c_u(v) = t \tag{4.5}$$

Here c_u is the partial derivative of $C(u, v)$ with respect to u. To calculate the value of v we construct the inverse of this partial derivative. The results we obtain are the values u and v, which are our sample values. These sample values we can compare to the pseudo observations of the historical data.

From Figure 4.4 we may notice that the Checkerboard copula fits well between the Fréchet-Hoeffding lower and upper bound, just as every other copula that exists. Also we notice that the planes are slightly curved.

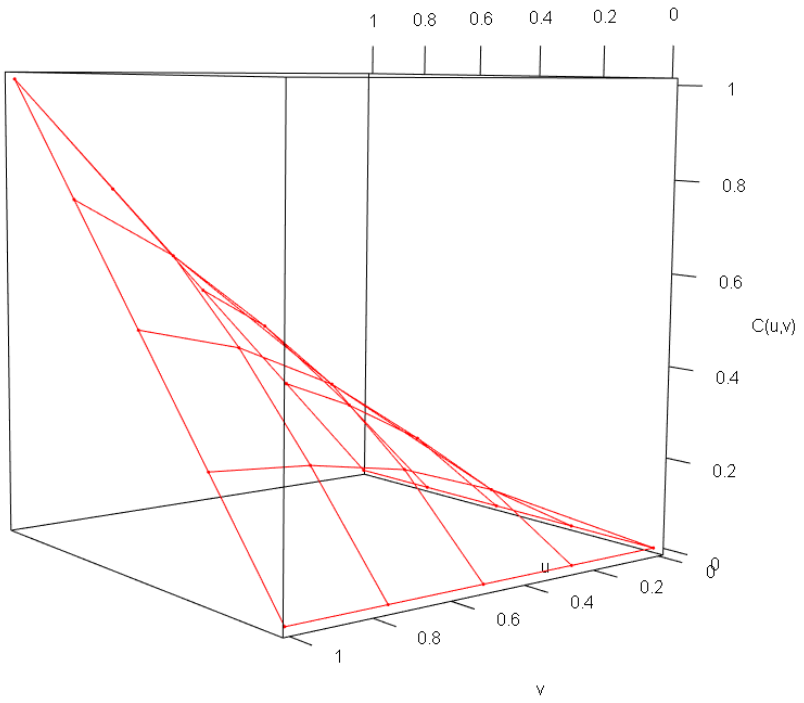


Figure 4.4: The checkerboard copula en profil

4.3.1 Comparing simulated data with the historical data

In Figure 4.5 we can see the result of our case study, a scatter plot of $5,000$ observations simulated with the checkerboard copula. To compare this figure with the historical data, we made pseudo observations of the historical data using the so-called 'pobs' function in R [10]. Pseudo observations are made by ranking all observations, and then dividing them into the unit square according to their rank. This result will be scaled by $\frac{n}{n+1}$, then the result will be our pseudo observations. I.e. if we have sixty observations, then the observation with the smallest u-value and the 30^{th} smallest v-value, then our pseudo observation will have a u-value of $\frac{1}{61}$, and a v-value of $\frac{30}{61}$. The pseudo observations can be seen in Figure 4.6, notice that it is similar to Figure 4.1. When we compare these to figures we notice similarities in the density of the partitions separated by the grid. Theoretically, the only connection between the two figures is that the correlation, Spearman's rho coefficient, is equal. However, when we compare the two figures we conclude that the simulated data seems very similar to the pseudo observations. Therefore the simulated data could be used if the given data does not contain enough observations. This could be useful when we would like to perform a model fit for the data. If we would like to create predictions for the rainfall.
After simulating we wanted to check that the Spearman rho value of the simulated data was 0.426, which was the value that we expected it should be as one of the constraints of the optimization problem. To take a better look at this we ran $10,000$ simulations and made a histogram of the Spearman rho values, which can be seen in Figure 4.7. We concluded that the mean value for the Spearman rho was 0.419, so it is slightly lower than the original constraint.

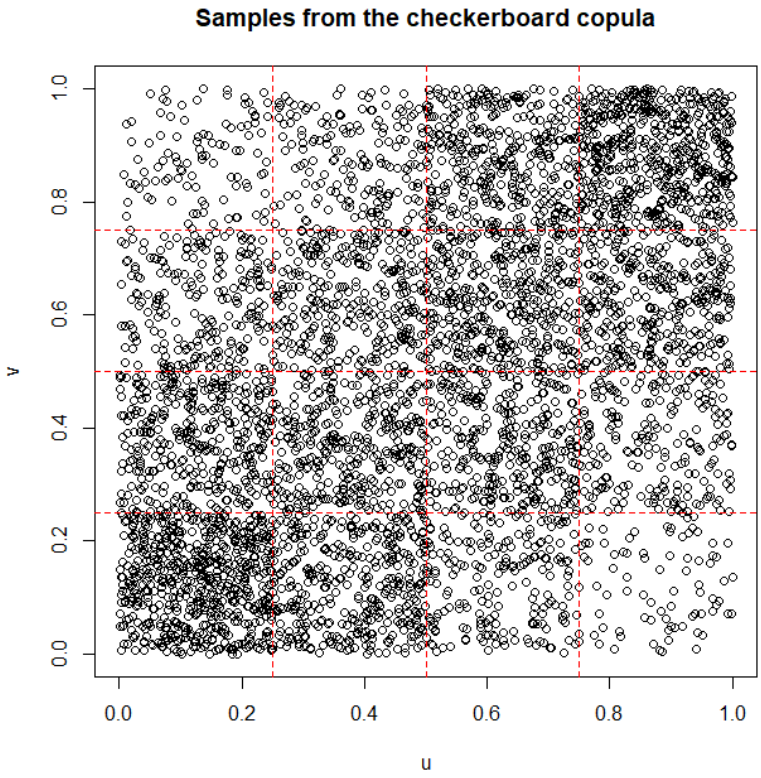


Figure 4.5: Scatter plot of the sampled values, with a grid

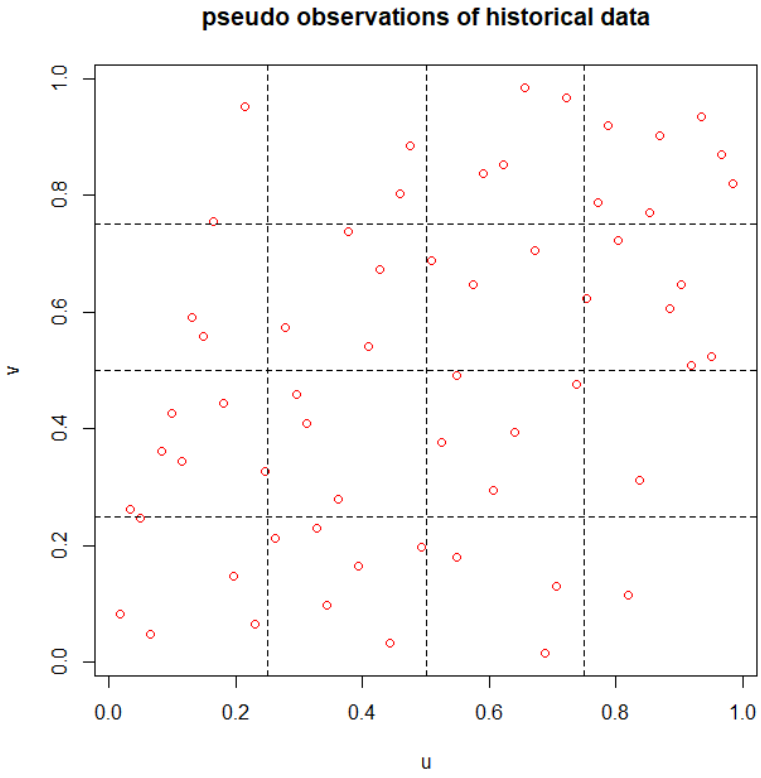


Figure 4.6: Scatter plot of the pseudo observations from the historical data

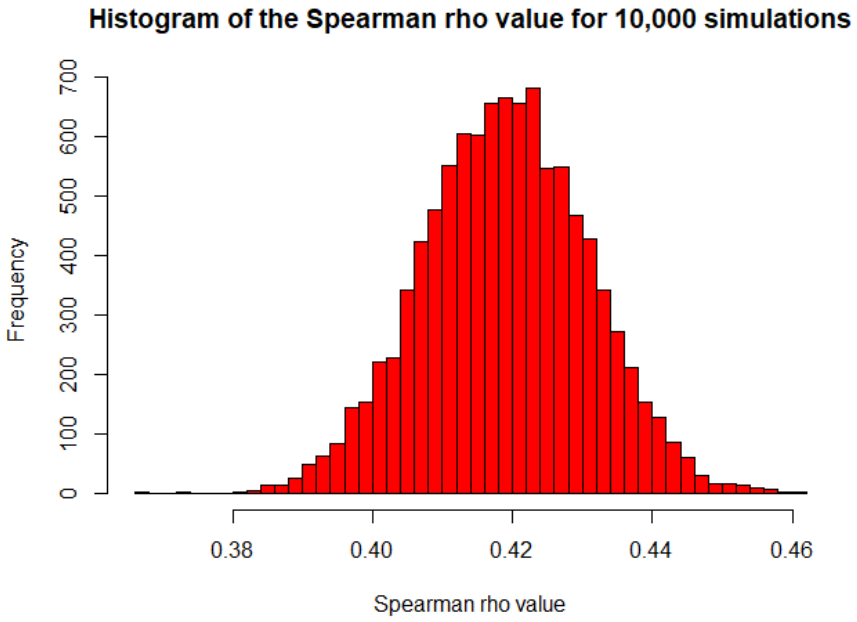


Figure 4.7: Histogram of the Spearman rho value for 10,000 different simulations of 10,000 observations

5 Conclusion

In the thesis we introduced copulas in general, then discrete copulas, and from there we discussed checkerboard copulas. We discussed a case study of the checkerboard copula approach to simulating data for rainfall totals in Eindhoven for January and September. In the case study we first constructed a doubly-stochastic matrix using an optimization problem. With this doubly-stochastic matrix, we constructed a discrete copula, then we extended the discrete copula to a full domain checkerboard copula. This checkerboard copula has maximum entropy and corresponds to our Spearman's rho value. At last, we made simulations with this checkerboard copula. These simulations have a significant resemblance with the historical data after we create pseudo observations of this historical data. Therefore we conclude the checkerboard copula model resembles the shape of the pseudo observations from the historical data.

With this thesis, we hope to make a small contribution to the checkerboard copula field by summarizing ways of constructing checkerboard copulas and creating simulations with it. Also, it is possible to create a more general code for simulating with checkerboard copulas, such that it works for all different sizes of checkerboard copulas, and not just for those that are 4×4. We would like to note that the simulated data seemed to consistently have a slightly lower Spearman's rho value, thus the data was a bit less correlated. We are not sure what is causing this, therefore we also suggest looking at this for future work. At last we suggest a comparisons with other copula models for future work.

Acknowledgments

We are very grateful to Professor Uryasev, co-author of the article "Checkerboard copulas defined by sums of random variables" [14] for his guidance and help with solving the optimization problem in PSG. Then we would like to thank Emma Zwerus and Jerom van der Zande for proof reading the introduction and giving feedback. We thank Hester de Jongh for proof reading the thesis and helping with the issues overleaf/latex presented. Our gratitude goes to Sarah Zwerus for the design of the figure after the title page, her art is inspiring as always. Last but not least we thank Dr. Perrone for supervising the thesis, giving lots of feedback, and sticking with the project even though it might have looked unpromising at times.

Bibliography

[1] J. Borwein and P. Howlett. “CHECKERBOARD COPULAS OF MAXIMUM ENTROPY WITH PRESCRIBED MIXED MOMENTS”. In: *Journal of the Australian Mathematical Society* 107 (2018), pp. 302–318.

[2] American optimal deceisions inc. *PSG software*. URL: http://www.aorda.com/html/PSG_Help_HTML/index.html?introduction_to_psg.htm.

[3] A. Kolesarova, R. Mesiar, J. Mordelova, and C. Sempi. “Discrete Copulas”. In: *IEEE Transactions on Fuzzy Systems* 14.5 (2006), pp. 698–705. DOI: 10.1109/TFUZZ.2006.880003.

[4] MATLAB. *version 7.10.0 (R2010a)*. Natick, Massachusetts: The MathWorks Inc., 2010.

[5] Radko Mesiar. “Discrete copulas - what they are.” In: *Fuzzy Logic and Technology* (Jan. 2005), pp. 927–930.

[6] Roger Nelsen. *An introduction to Copulas*. 2nd edition. Springer, 2006.

[7] World meteorological organization. URL: http://climexp.knmi.nl/data/hom1951902_sum12_anom.dat.

[8] Julia Piantadosi, Phil Howlett, and John Boland. “Matching the grade correlation coefficient using a copula with maximum disorder”. In: *Journal of Industrial and Management Optimization* 3 (May 2007). DOI: 10.3934/jimo.2007.3.305.

[9] Julia Piantadosi, Phil Howlett, and Jonathan (Jon) Borwein. “Copulas with Maximum Entropy”. In: *Optimization Letters* 6 (Jan. 2012), pp. 99–125. DOI: 10.1007/s11590-010-0254-2.

[10] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2018. URL: https://www.R-project.org/.

[11] Goovaerts Rob Kaas. *Modern actuarial risk theory*. 2nd edition. Springer, 2008.

[12] M. Sklar. “Fonctions de répartition a n dimensions et leurs marges”. In: *Publications de l’Institut Statistique de l’Université de Paris* (1959).

[13] Pravin K. Trivedi and David M. Zimmer. “Copula Modeling: An Introduction for Practitioners”. In: *Foundations and Trends® in Econometrics* 1.1 (2007), pp. 1–111. ISSN: 1551-3076. DOI: 10.1561/0800000005. URL: http://dx.doi.org/10.1561/0800000005.

[14] Kuzmenko Viktor, Salam Romel, and Uryasev Stan. “Checkerboard copula defined by sums of random variables”. In: *Dependence Modeling* (2020), pp. 70–92. URL: https://EconPapers.repec.org/RePEc:vrs:demode:v:8:y:2020:i:1:p:70-92:n:4.

Appendix

Simulation

```
library ( plotly )
library (copula)
data ← read.delim("dataRainEindSum.txt", sep = "")
df = as.data.frame(mydata)
plot_ly(df, x = Y, y = X, z = Z, group = X, type = "scatter3d", mode = "lines")

a ← c(4.684144847744632E−01
,3.025637932214215E−01
,1.584885639690427E−01
,7.053315790210214E−02
,2.986847774797829E−01
,2.935893778476683E−01
,2.492372860296781E−01
,1.584885636828886E−01
,1.618118368500140E−01
,2.420349855247981E−01
,2.935893784632549E−01
,3.025637933095406E−01
,7.108890415410679E−02
,1.618118367579934E−01
,2.986847779361946E−01
,4.684144846445030E−01) #Assign values of doubly− stochastic matrix (in an array)


#−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
p← a[1]/4
b ← (a[1]+a[2])/4
c ← (a[1]+a[2]+a[3])/4
d ← (a[1]+a[5])/4
e ← (a[1]+a[2]+a[5]+a[6])/4
f ← (a[1]+a[2]+a[3]+a[5]+a[6]+a[7])/4
g ← (a[1]+a[5]+a[9])/4
h ← (a[1]+a[2]+a[5]+a[6]+a[9]+a[10])/4
i ← (a[1]+a[2]+a[3]+a[5]+a[6]+a[7]+a[9]+a[10]+a[11])/4 #Create values of discrete copula using the procedure from proposition 1

i

DC← matrix(c(p,b,c,0.25,d,e,f,0.5,g,h,i ,0.75,0.25,0.5,0.75,1) , ncol = 4) #Discrete copula
DC


#DC ← matrix(c(p,d,g,0.25,b,e,h,0.5,c,f,i ,0.75,0.25,0.5,0.75,1) ,nrow = 4)


#−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

sample ← function(DC){
  v ← rep(0,5000)
  u ← runif(5000,0,1)
  t ← runif(5000,0,1)
  for(i in 1:5000){
    if(u[i]<0.25){
      if(t[i]<0.4684145){
        v[i]← t[i]/(16*DC[1,1])
      }else if (t[i]<0.7670993){
        v[i]← (t[i]−8*DC[1,1]+4*DC[1,2])/(16*(DC[1,2]−DC[1,1]))
      }else if (t[i]< 0.9289111){
        v[i]←(t[i]−12*DC[1,2]+8*DC[1,3])/(16*(DC[1,3]−DC[1,2]))
      }else{
        v[i]←(t[i]+3−16*DC[1,3])/(4−16*DC[1,3])
      }
    }else if(u[i]<0.5){
      if(t[i]<0.3025638){
        v[i]←t[i]/(16*DC[2,1]−16*DC[1,1])
      }else if(t[i]<0.5961532){
        v[i]← (t[i]−4*(−2*DC[1,1]+DC[1,2]+2*DC[2,1]−DC[2,2]))/(16*(DC[1,1]−DC[1,2]−DC[2,1]+DC[2,2]))
      }else if(t[i]<0.8381882){
        v[i]← (t[i]−4*(−3*DC[1,2]+2*DC[1,3]+3*DC[2,2]−2*DC[2,3]))/(16*(DC[1,2]−DC[2,2]−DC[1,3]+DC[2,3]))
      }else{
        v[i]←(t[i]−16*(DC[2,3]−DC[1,3])+3)/(4*(4*DC[1,3]−4*DC[2,3]+1))
```

```
        }
    } else if (u[i]<0.75){
        if (t[i]<0.1584886){
            v[i]<-t[i]/(16*(DC[3,1]-DC[2,1]))
        } else if (t[i]<0.4077258){
            v[i]<-(t[i]-4*(-2*DC[2,1]+DC[2,2]+2*DC[3,1]-DC[3,2]))/(16*(DC[2,1]-DC[2,2]-DC[3,1]+DC[3,2]))
        } else if (t[i]<0.7013152){
            v[i]<-(t[i]-4*(-3*DC[2,2]+2*DC[2,3]+3*DC[3,2]-2*DC[3,3]))/(16*(DC[2,2]-DC[2,3]-DC[3,2]+DC[3,3]))
        } else {
            v[i]<-(t[i]-16*(DC[3,3]-DC[2,3])+3)/(4*(4*DC[2,3]-4*DC[3,3]+1))
        }
    } else {
        if (t[i]<0.07053316){
            v[i]<- (t[i]/(4-16*DC[3,1]))
        } else if (t[i]<0.2290217){
            v[i]<- (t[i]-4*(-2*DC[3,1]+DC[3,2]))/(4*(4*DC[3,1]-4*DC[3,2]+1))
        } else if (t[i]<0.5315855){
            v[i]<-(t[i]-4*(2*DC[3,3]-3*DC[3,2]))/(4*(4*DC[3,2]-4*DC[3,3]+1))
        } else {
            v[i]<- (t[i]-9+16*DC[3,3])/(8*(2*DC[3,3]-1))
        }
    }
  }
  return (c(u,v))
}#Function that samples values for u and v, like Nelsen 2.9 explains, and lemma 2.3.5 from Nelsen

samples <- sample(matrix(c(p,b,c,0.25, d,e,f,0.5, g,h,i,0.75,0.25,0.5,0.75,1) , ncol = 4))
samples[20000]

u1<- samples[1:5000]
v1<- samples[5001:10000]
v1[v1<0.01]
v2<-v1[v1<=0.751]
v2[0.75<v2]
v3<-v1[1:250]
v1[v1<0]

plot(u1,v1, main = 'Samples from the checkerboard copula', xlab = 'u', ylab = 'v') #Plot the sampled values
#abline(lm(v1~u1), col = 'black')
abline (0.25,0, col='black', lty=2)
abline (0.5,0, col='black', lty=2)
abline (0.75,0, col='black', lty=2)
abline (v=0.25,col='black', lty=2)
abline (v=0.5,col='black', lty=2)
abline (v=0.75,col='black', lty=2) #add grid

data <- read.delim("dataRainEindSum.txt", sep = "") #load historical data of monthly totals of rainfall in Eindhoven
data
mean(data)

data$jan
length(data$sep)
data1<- matrix(c(data$jan,data$sep), ncol=2)
data1
data2 <-pobs(data1)#create pseudo observations of historical data

plot(data2, col='red', xlab='u', ylab ='v', main= 'pseudo observations of historical data')


lm(v1~u1)

16*b*DC[1,1]
b<-0.25
16*b*(DC[1,2]-DC[1,1])+8*DC[1,1]-4*DC[1,2]
b<- 3/4
16*b*(DC[1,3]-DC[1,2])+12*DC[1,2]-8*DC[1,3]
b<- 1
4*b-16*b*DC[1,3]+16*DC[1,3]-3

(16*DC[2,1]-16*DC[1,1])*b
b<-0.5
16*b*(DC[1,1]-DC[1,2]-DC[2,1]+DC[2,2])+4*(-2*DC[1,1]+DC[1,2]+2*DC[2,1]-DC[2,2])
16*b*(DC[1,2]-DC[1,3]-DC[2,2]+DC[2,3])+4*(-3*DC[1,2]+2*DC[1,3]+3*DC[2,2]-2*DC[2,3])
4*b*(4*DC[1,3]-4*DC[2,3]+1)+16*(DC[2,3]-DC[1,3])-3

16*b*(DC[3,1]-DC[2,1])
```

```
16*b*(DC[2,1]-DC[2,2]-DC[3,1]+DC[3,2])+4*(-2*DC[2,1]+DC[2,2]+2*DC[3,1]-DC[3,2])
16*b*(DC[2,2]-DC[2,3]-DC[3,2]+DC[3,3])+4*(-3*DC[2,2]+2*DC[2,3]+3*DC[3,2]-2*DC[3,3])
4*b*(4*DC[2,3]-4*DC[3,3]+1)+16*(DC[3,3]-DC[2,3])-3

-16*b*DC[3,1]+4*b
4*b*(4*DC[3,1]-4*DC[3,2]+1)+4*(-2*DC[3,1]+DC[3,2])
4*b*(4*DC[3,2]-4*DC[3,3]+1)+4*(2*DC[3,3]-3*DC[3,2])
8*b*(2*DC[3,3]-1)+9-16*DC[3,3] #These are the partial derivatives which I used to check if they are right, and their inverses are right

cor.test(u1, v1,
         method = "spearman")

samples2 <- matrix(rep(0,100000000),nrow = 10000)
u2 <- matrix(rep(0,50000000),nrow = 10000)
v2 <- matrix(rep(0,50000000),nrow = 10000)
SP_rho <- rep(0,10000)
SP_rho_est <- rep(0,10000)
typeof(SP_rho)

for (i in 1:10000){
  samples2[i,] <-sample(DC)
  u2[i,] <-samples2[i,1:5000]
  v2[i,] <-samples2[i,5001:10000]
  cor2 <- cor.test(u2[i,], v2[i,])

  SP_rho_est[i] <- cor2$estimate
}

SP_rho_est
hist(SP_rho_est, breaks = 50,col = 'red', xlab = 'Spearman rho value', main = 'Histogram of the Spearman rho value for 10,000 simulations')
summary(SP_rho_est)

#-----------------------------------------------------------------------

library(rgl)
# Create some dummy data
dat <- replicate(2, 1:3)
dat



plot3d(dat, type = 'n', xlab = 'u', ylab = 'v', zlab = 'C(u,v)', xlim = c(1,0),ylim= c(1,0),zlim= c(1,0))

points3d(0.25,0.25, DC[1,1], col='red')
points3d(0.25,0.5, DC[2,1], col='red')
points3d(0.25,0.75, DC[3,1], col='red')
points3d(0.25,1, DC[4,1], col='red')

points3d(0.5,0.25, DC[1,2], col='red')
points3d(0.5,0.5, DC[2,2], col='red')
points3d(0.5,0.75, DC[3,2], col='red')
points3d(0.5,1, DC[4,2], col='red')

points3d(0.75,0.25, DC[1,3], col='red')
points3d(0.75,0.5, DC[2,3], col='red')
points3d(0.75,0.75, DC[3,3], col='red')
points3d(0.75,1, DC[4,3], col='red')

points3d(1,0.25, DC[1,4], col='red')
points3d(1,0.5, DC[2,4], col='red')
points3d(1,0.75, DC[3,4], col='red')
points3d(1,1, DC[4,4], col='red')

points3d(0,0,0, col='red')
points3d(0.25,0,0, col='red')
points3d(0.5,0,0, col='red')
points3d(0.75,0,0, col='red')
points3d(1,0,0, col='red')
points3d(0,0.25,0, col='red')
points3d(0,0.5,0, col='red')
points3d(0,0.75,0, col='red')
points3d(0,1,0, col='red') #Here we draw the points that make up the discrete copula


```

```
lines3d(c(0,0.25),c(0,0.25),c(0,DC[1,1]),col='red')
lines3d(c(0,1),0,0,col='red')
lines3d(0,c(0,1),0,col='red')
lines3d(c(0,0.25),0.25,c(0,DC[1,1]),col='red')
lines3d(0.25,c(0,0.25),c(0,DC[1,1]),col='red')
lines3d(0.5,c(0,0.25),c(0,DC[1,2]),col='red')
lines3d(0.75,c(0,0.25),c(0,DC[1,3]),col='red')
lines3d(1,c(0,0.25),c(0,DC[1,4]),col='red')
lines3d(c(0.25,0.5),0.25,c(DC[1,1],DC[1,2]),col='red')
lines3d(c(0.5,0.75),0.25,c(DC[1,2],DC[1,3]),col='red')
lines3d(c(0.75,1),0.25,c(DC[1,3],DC[1,4]),col='red')
lines3d(0.25,c(0.25,0.5),c(DC[1,1],DC[2,1]),col='red')
lines3d(0.25,c(0.5,0.75),c(DC[2,1],DC[3,1]),col='red')
lines3d(0.25,c(0.75,1),c(DC[3,1],DC[4,1]),col='red')
lines3d(c(0,0.25),0.5,c(0,DC[2,1]),col='red')
lines3d(c(0,0.25),0.75,c(0,DC[3,1]),col='red')
lines3d(c(0,0.25),1,c(0,DC[4,1]),col='red')
lines3d(c(0.25,0.5),c(0.25,0.5),c(DC[1,1],DC[2,2]),col='red')
lines3d(c(0.5,0.75),c(0.5,0.75),c(DC[2,2],DC[3,3]),col='red')
lines3d(c(0.75,1),c(0.75,1),c(DC[3,3],DC[4,4]),col='red')
lines3d(0.5,c(0.25,0.5),c(DC[1,2],DC[2,2]),col='red')
lines3d(c(0.25,0.5),0.5,c(DC[2,1],DC[2,2]),col='red')

lines3d(0.75,c(0.25,0.5),c(DC[1,3],DC[2,3]),col='red')
lines3d(1,c(0.25,0.5),c(DC[1,4],DC[2,4]),col='red')
lines3d(0.75,c(0.5,0.75),c(DC[2,3],DC[3,3]),col='red')
lines3d(1,c(0.5,0.75),c(DC[2,4],DC[3,4]),col='red')
lines3d(1,c(0.75,1),c(DC[3,4],DC[4,4]),col='red')
lines3d(c(0.5,0.75),0.5,c(DC[2,2],DC[2,3]),col='red')
lines3d(c(0.75,1),0.5,c(DC[2,3],DC[2,4]),col='red')
lines3d(c(0.75,1),0.75,c(DC[3,3],DC[3,4]),col='red')
lines3d(c(0.25,0.5),0.75,c(DC[3,1],DC[3,2]),col='red')
lines3d(c(0.5,0.75),0.75,c(DC[3,2],DC[3,3]),col='red')
lines3d(c(0.25,0.5),1,c(DC[4,1],DC[4,2]),col='red')
lines3d(c(0.5,0.75),1,c(DC[4,2],DC[4,3]),col='red')
lines3d(c(0.75,1),1,c(DC[4,3],DC[4,4]),col='red')
lines3d(0.5,c(0.5,0.75),c(DC[2,2],DC[3,2]),col='red')
lines3d(0.5,c(0.75,1),c(DC[3,2],DC[4,2]),col='red')
lines3d(0.75,c(0.75,1),c(DC[3,3],DC[4,3]),col='red') #Here we draw the lines that make the Checkerboard copula
```