Johan Øvstebø Birketvedt

Interval Censored Regular Vines With Application to Event-Based Modelling of Precipitation and Temperature

Masteroppgave i Fysikk og matematikk
Veileder: Sara Martino
Juni 2019

Masteroppgave

NTNU
Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for matematiske fag

NTNU
Kunnskap for en bedre verden

Johan Øvstebø Birketvedt

Interval Censored Regular Vines With Application to Event-Based Modelling of Precipitation and Temperature

Masteroppgave i Fysikk og matematikk
Veileder: Sara Martino
Juni 2019

Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for matematiske fag

NTNU
Kunnskap for en bedre verden

Abstract

Many types of extreme weather are associated with the dependence of precipitation and temperature. In susceptible regions, periods of high temperature and low precipitation cause drought, while periods of high precipitation and high temperature are associated with floods in mountain regions, as the high temperatures increase glacial drainage. In this work we investigate the relationship by the precipitation events model of De Michele & Salvadori (2003), to which we include a measure for temperature. All parameters are modeled jointly by multivariate vine copulae, or pair-copula-constructions, which is a flexible tool for modelling non-Gaussian multivariate distributions. Typical data for hydrological studies consist of hourly measurements of precipitation and temperature. These contain many duplicated measurements (ties), in particular in the lower tails, which is a source of bias in the widely used rank-based estimation methods. Bivariate interval censored estimation was shown in Li et al. (2016) to be unbiased in the presence of ties. Two methods are proposed to extend interval censoring to multivariate vine copulae, and these are tested in a large scale simulation study. The methods are unbiased in the low levels the vine, but not generally in higher trees when correlations are strong. The best performing method, denoted full censoring, still shows some improvements in these cases, and is emphasized in the application to events. Precipitation events of the model in De Michele & Salvadori (2003) are assumed to be i.i.d in each season. However, temperature has a clear seasonal trend, and since the events form an irregular time series, the parameter is modelled by Fourier terms with ARIMA correction. The available data for this study is of low quality, and the estimated dependence is weaker than expected, so the full precipitation-temperature modelling of events is more a conceptual demonstration of interval censored regular vines. We construct one larger weather model to demonstrate structural differences in each season, and one smaller model to emphasize the relationship between precipitation intensity, duration and temperature.

Sammendrag

Mange typer ekstremvær skyldes årsaker som følge av sammenhengen mellom temperatur og nedbør. I utsatte områder forårsaker perioder med høy temperatur og lite nedbør tørke, mens perioder med høy temperatur og nedbør er assosiert med flom i fjellområder, grunnet økning i smeltevann fra isbreer. I denne oppgaven ser vi på denne sammenhengen ved å utvide bygemodellen fra De Michele & Salvadori (2003) til å inkludere temperatur. Alle modell-parameterne kan modelleres i en simultanfordeling ved bruk av par-copula-konstruksjoner, som er et generelt verktøy for modellering av ikke-Gaussiske multivariate fordelinger. Typiske værdata består av timesmålinger av nedbør og temperatur. Disse inneholder mange duplikater, spesielt i de lave halene, som bidrar til forventningsskjevhet i estimasjonene av kontinuerlige prosesser. Li et al. (2016) viste at så kalt "intervall-sensurert" estimatsjon er forventningsrett i to dimensjoner. Vi foreslår to metoder for å generalisere intervall-sensurering til multivariate par-copula-konstruksjoner, og disse blir testet i en simuleringsstudie. Metodene er forventningsrette i lave nivåer av par-copula-konstruksjonene, men ikke generelt i de høye nivåene når korrelasjonene er sterke. Den beste metoden er fortsatt noe bedre enn alternativene, og blir fokuset for anvendelsen til bygemodellen. I modellen fra De Michele & Salvadori (2003) er observasjonene antatt å være uavhengige realiseringer fra samme fordeling innad i hver sesong. Temperatur har, imidlertid, en klar sesongbasert trend, og siden bygeobservasjonene er en irregulær tidsrekke, blir det brukt Fourier-ledd med ARIMA-korreksjon for å modellere temperatur. De tilgjengelige dataene for denne studien er av lav kvalitet, og den estimerte avhengigheten mellom temperatur og nedbør er lavere enn forventet, så den fulle temperatur-nedbørsmodellen er heller en konseptuell demonstrasjon av intervall-sensurerte regulære par-copula-konstruksjoner. Vi lager en større modell, med fokus på ulike konstruksjonsstrukturer for hver sesong, og en mindre som mer spesifikt modellerer forholdet mellom nedbørintensitet, varighet og temperatur.

Preface

This master thesis was written in completion of my Master of Science in Physics and Mathematics at the Norwegian University of Science and Technology (NTNU). Throughout the study, I have have explored a wide range of topics ranging from statistical modelling to cryptocurrencies and machine learning. This is was in part made possible by academic exchanges to The University of New South Wales (UNSW), Sydney, and to Tokyo Institute of Technology (東京工業大学).

During the fall semester of 2018, we decided to learn about a new topic – copula, and apply the theory on bivariate data. This thesis is a continuation of this work. Here we wish to explore the possibilities for multivariate modelling to build on the previous models. I would like to thank my supervisor, Assoc. Prof. Sara Martino, for the continued support. She has provided solid literature on the topics, and assisted in structuring the study. Her advice has always been helpful.

Table of Contents

Chapter 1

Introduction

The relationship between precipitation and temperature is particularly interesting, as it is associated with extreme weather. In susceptible regions, long periods of high temperature and low precipitation cause drought. Furthermore, mountain region floods are connected to high precipitation and high temperature due to the increased glacial drainage. In this thesis, we want to model the relationship by precipitation events. Informally, an event is a period of rain, followed by a by a minimum dry period, i.e. 5 hours. Each event is typically characterized by the event parameters precipitation volume V, duration W, mean intensity $I = V/W$ and the length of the preceding dry period D. This model does not typically include temperature.

According to the Calusius-Clapeyron rate, the water vapor holding capacity increases with air temperature at a rate of approximately $7\%^{\circ}\mathrm{C}^{-1}$, which is expected to cause an increase in the precipitation intensity (Panthou et al., 2014). This motivates the construction of a multivariate event model with the inclusion of temperature. Other works that aim to model precipitation and temperature generally measure correlation on a larger time scale, i.e. by looking at monthly or daily means, as in Lenderink & van Meijgaard (2008); Panthou et al. (2014); Molnar et al. (2015). In Panthou et al. (2014) and Molnar et al. (2015), the authors also use a similar event model, however, mainly to find the rate which intensity increases with temperature, and not a joint model. Extreme compound events are in short, events where each contributing factor in it self is not extreme, but jointly, they produce an extreme compound event, i.e. drought or floods. A full multivariate model can be used to quantify risk of extreme compound events.

The stochastic modelling of the event parameters has previously been difficult, and was done under the assumption of multivariate independence between the intensity and duration of an event. Storms were described as rectangular pulses with an arrival rate following a Poisson process (Salvadori & Michele, 2007). Independence was assumed as a consequence of modelling difficulty of non-Gaussian multivariate models, and has later been lifted following advancements in the field by De Michele & Salvadori (2003); Salvadori & Michele (2007). The authors introduce a precipitation event model which exploits the theory of copula, and allows for a separate modelling of the joint and marginal distri-

butions. This was the motivation behind the authour's previous work Birketvedt (2019), where the relationship between precipitation intensity and duration at Risvollan, Trondheim, Norway, was modelled using copulae. To the authors knowledge, temperature and precipitation has not been modelled with an event model using copula. This motivates the extension of the model from Birketvedt (2019) to include temperature.

Hydrological applications of copulae have mostly been restricted to the bivariate case (De Michele & Salvadori, 2003; Vandenberghe et al., 2010). Bivariate copulae are heavily researched, and there exist few of larger dimension. In Aas et al. (2009), the authors introduce the vine copula which allows for a construction of a large multivariate copula by combinations of bivariate copulae. Advancements in model construction and simplification were later made in Brechmann et al. (2012); Dißmann et al. (2012). Vines, or pair-copula constructions, have since been applied by Bevacqua et al. (2017) to risk assessment on extreme compound flood events. Due to the general applicability of vines, this seems like a natural choice for constructing a larger model for precipitation events also.

The intended topic for this thesis was to apply vines to build a multivariate model for precipitation events, and investigate the different dependence structures of intensity, duration and temperature across Norway. The data available was ultimately of insufficient quality for this purpose. An underlying assumption in copula modelling is that the processes are continuous, and do not have duplicated measurements (ties). In Norway, precipitation is most commonly measured over intervals of 1 hour, which will effectively cause a rounding error in the true precipitation. The 1 hour interval is large to the extent where the rounding severely affects the measurement precision, and we get many ties. This issue is rarely mentioned in the literature. In Salvadori & Michele (2006); Panthou et al. (2014) ties are managed by introducing lower threshold for precipitation volume in each event, and in Vandenberghe et al. (2010) they introduce random noise to duplicated measurements. Neither of these approaches are perfect as they either do not sufficiently account for ties, or cause estimation bias. In Li et al. (2016) the authors propose a solution for bivariate models, and introduce the concept of interval censoring, which has shown signs of unbiased estimation in the presence of ties. In this study, we attempt to construct interval censored vines, and demonstrate a use case on a precipitation event model which includes temperature. We also provide some methods for constructing precipitation events that account for temperature.

Chapter 2

The Dataset

We collected data about precipitation and temperature from 32 weather stations owned by the Norwegian Meteorological Institute. The time series span the period from January 1st 1983 to December 31st 2018, and contains hourly measurements of precipitation and temperature. The overall quality of data is varying across all weather stations, and most stations have a large amount of missing data. Out of all measurements 9% are missing for temperature, and 39% for precipitation. We want to select one station for this study. To do this, we first define an event in Section 2.1, and in Section 2.2 we select one station that present the highest number of events. In Section 2.2 the selected station is investigated. Ties cause bias in the modelling of events, and the severity of the issue is quantified. It turns that the data quality is a larger problem than expected.

2.1 Event Definition

A precipitation event (or storm) is defined as a rainy period, separated from the next rainy period by a defined number of dry hours. Precipitation events are characterized by the joint behaviour of several random variables such as volume V, that is the total amount of precipitation recorded during the event, the event duration W, the mean intensity $I = V/W$ and the dry period D preceding the event itself. Figure 2.1 shows an illustration of these parameters. The events model is widely used in the hydrological literature, i.e. De Michele & Salvadori (2003); Salvadori & Michele (2007); Vandenberghe et al. (2010); Panthou et al. (2014); Molnar et al. (2015), and was used in the authors previous work Birketvedt (2019). In the literature, typical choices for the dry separation length range from 5 hours to 24 hours (Vandenberghe et al., 2010; De Michele & Salvadori, 2003; Birketvedt, 2019). Here we have chosen the separation length to be 5 hours. The models do not currently have a suggested parameter for temperature, so we measure eight additional candidate temperature parameters. During the event, we measure the mean temperature T, maximum temperature T_M, minimum temperature T_m and the maximum temperature difference T_Δ, and during the dry period we measure mean temperature T_D, maximum temperature T_{DM}, minimum temperature T_{Dm} and the maximum temperature difference $T_{D\Delta}$. The precipi-

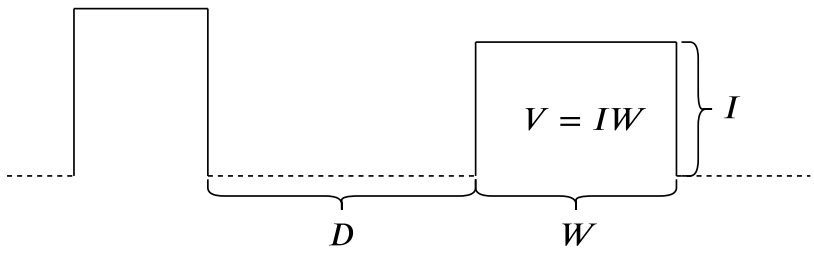
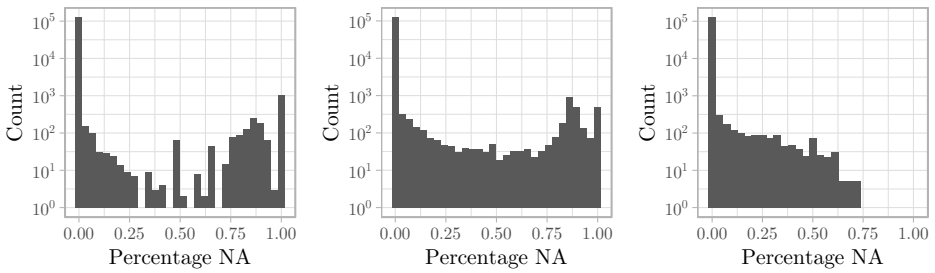


Figure 2.1: Illustration of the parameters in the standard event model. Each rectangular pulse denotes an event.

tation is measured by a tipping bucket with a typical measurement resolution of 0.1 mm. Each time the bucket is filled, its contents are tipped into a larger container. The hourly rain volume is typically calculated from the number of tips. In some cases, the measurements are negative, which is mostly assumed to be caused by evaporation. All negative measurements are labeled as *missing* (NA). There are also outliers present. Hence, the measurements across all stations are filtered from the most relevant records gathered from *Norgesrekorder* (n.d.); *Regnværet setter nye rekorder* (2014), which give a reasonable indication on the parameter bounds.

Each event is separated by a dry period of 5 hours, and in this regard, missing measurements are treated as dry periods. This means that the 5 hour separation period can consist of both dry and/or missing measurements. The remaining parameters are calculated from the relevant non-missing values for each event. That is, V and I are computed by simply removing missing values, i.e. the series $(1, 1, \mathrm{NA}, 1)$ would have $V = 3$ and $I = 1$. Similarly, the temperature parameters are computed from only complete measurements. For W, we compute the total duration, even for missing rain. Thus, the same series as before would have $W = 4$. Following this approach, event parameters could potentially be calculated from a large amount of missing measurements. There is the possibility that an event with duration $W = 96$ could consist of volume measurements $V = \{$ 0.1, NA, NA, NA, NA, 0.1, ..., NA, 0.1$\}$, such that the parameters V, I would be calculated from only 20 complete measurements. Figure 2.2 shows histograms of the amount of missing measurements within each event for the parameters mean temperature T, mean dry period temperature T_D and rain volume V for all weather stations. It is apparent that all parameters are calculated from a significant amount of missing measurements, which suggests that this should be considered when selecting a weather station. There are a significant amount of events where rain volume V is calculated from over 50% missing measurements. The average dry temperature T_D is generally estimated with an even larger amount of missing measurements, and there are many events with up to 99% measurements missing. It should be noted that it is more crucial to have few missing measurements in rain volume V compared to the temperature parameters. Temperature is a much smoother process, so the uncertainty induced by missing measurements is smaller than for total rain volume V. Over 4 hours, large amounts of rain can accumulate, but the mean temperature is assumed to be relatively unchanged.

Following this event specification, around 20% of all events have a total volume $V =$



(a) T **(b)** T_D **(c)** V

Figure 2.2: Percentage NA measurements in all events for the parameters V, T and T_D selected following a dry period of 5 hours.

0.1 and duration $W = 1$. This can be considered noise, and it will cause estimation bias in copula modelling. Commonly, events are filtered by specifying a minimal value, or threshold, for one parameter, i.e. V or I (Salvadori & Michele, 2007, 2006; Panthou et al., 2014). We use $V > 1$ as a threshold for the events.

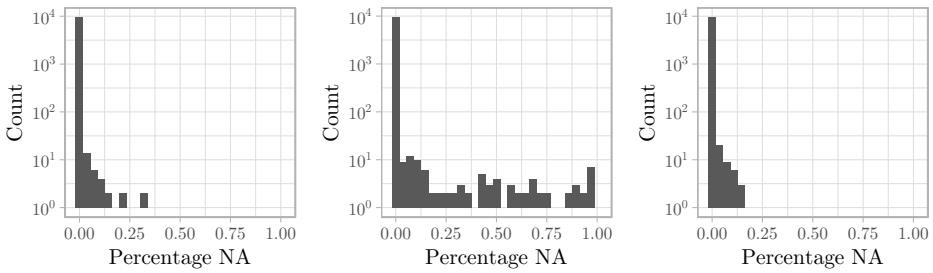
2.2 Særheim Weather Station

We want to restrict the analysis to one weather station with as many valid events as possible. This gives the station at Særheim, which is located in Rogaland in western Norway. There are a total of 3631 events, divided by seasons as shown in Table 2.1. The number of events for each season ranges from 847 in spring to 980 in summer.

Season	# Events
Winter	980
Spring	847
Summer	852
Fall	952
Total:	3631

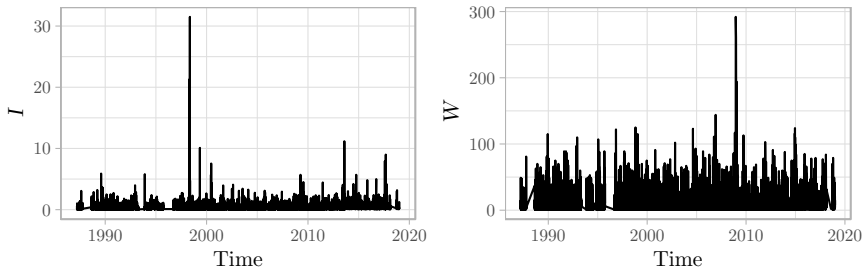
Table 2.1: Number events at Særheim divided by seasons. The events are selected following a dry period 5 hours.

While there are many valid events at Særheim, the parameters may still be calculated based on a large amount of missing measurement. As shown in Figure 2.3, the parameters V and T have fewer than 33% missing measurements for all events. The parameter T_D, on the other hand, has a significant amount of missing measurement, even close to 100%. This is not perfect, but still better than the dataset viewed as a whole. Figure 2.4 shows the observations of intensity I and duration W at Særheim. The mean intensity is 0.9 mm/h with a few large observations, whereas the mean duration is 19 h.



(a) T **(b)** T_D **(c)** V

Figure 2.3: Percentage NA measurements in events at Særheim for the parameters V, T and T_D selected following a dry period of 5 hours.



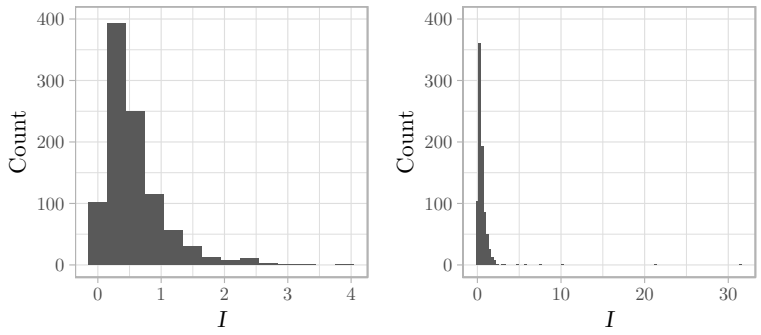
(a) I **(b)** W

Figure 2.4: Observations of the event parameters intensity I and duration W at Særheim following a dry period of 5 hours.

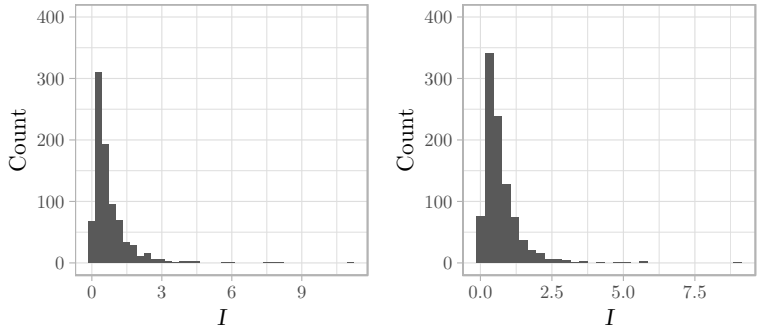
In the event model, the underlying multivariate distribution of the event parameters is assumed to be different for each season, hence, the events are divided into four seasons, where winter is the months December, January and February, spring is March, April and May, summer is June, July and August, and fall is September, October and November. The Figures 2.5 and 2.6 show histograms of I and W for each season. The intensity has the greatest range in spring, and the smallest in winter. The events are typically shortest in spring and summer, and longest in winter.

An underlying assumption when applying the theory of copula is that the marginal distributions are continuous, which means a *zero* probability of duplicated measurements (ties). However, even for continuous marginals, duplicates may still occur due to measurement imprecision. Table 2.2 shows the total amount of ties for I, W, V, D and T for each season, and for the entire dataset. The percentage of ties range from 5% for T during fall to 92.7% for D during spring. The duration W and volume V are typically around 70% and 90% respectively. In cases where the amount of ties is low, a common practice is to assume that it does not significantly affect the estimation and inference procedures, however, this cannot be considered reasonable for these data.

The data quality is in particular poor for the duration W. Table 2.3 shows the amount



(a) Winter **(b)** Spring



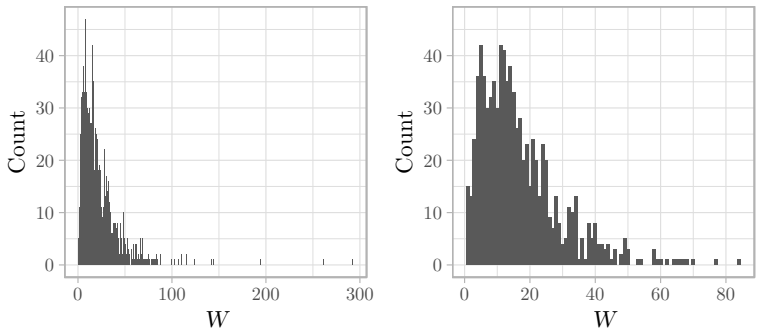
(c) Summer **(d)** Fall

Figure 2.5: Histogram of the mean precipitation intensity I for each season. The observations are counted in bins of size 0.3 mm/h.

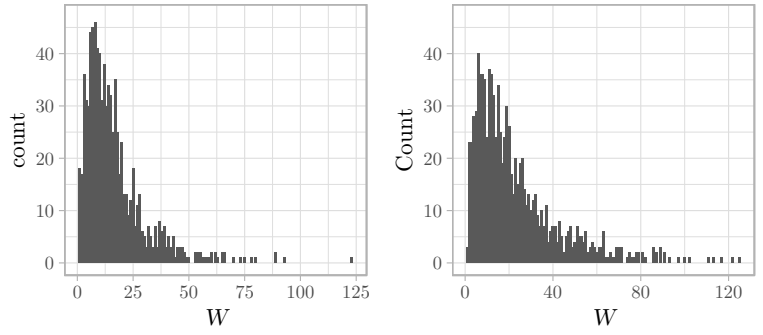
Season\# Ties	I	W	V	D	T
Winter	230	882	636	910	54
Spring	228	748	581	785	59
Summer	192	782	546	786	93
Fall	192	859	571	884	49
Total:	1593	3516	3061	3513	462

Table 2.2: Tied observations of the characteristic parameters intensity I, duration W, volume V, the dry period D and the mean temperature T for the different seasons based on a selection of events following a dry period D of 5 h. Total denotes the ties amongst the full 3631 events, and not the sum of ties for all seasons.

of ties for $W = 1 \ldots, 10$. For each hour, there are around 50 to 160 duplicates. This causes estimation bias, and in section 3.6.1 we will go into more detail regarding the challenges with common practices of treating ties, and discuss *interval censoring*, which



(a) Winter **(b)** Spring



(c) Summer **(d)** Fall

Figure 2.6: Histogram of the the precipitation duration W for each season. The observations are counted in bins of size 1 h.

is an unbiased estimation method in the presence of ties. The large number of ties is an indication of severe rounding error, and the true values are censored.

Duration W	Frequency
1	41
2	64
3	108
4	127
5	134
6	158
7	144
8	161
9	143
10	123
⋮	⋮

Table 2.3: Frequency of measurements for the duration W based on a selection of events following a dry period 5 hours. In a continuous process, all measurements should ideally have frequency 1. There are 3631 events in total.

Chapter 3

Theory

Copulae are multivariate probability distributions with uniform marginals on the interval $[0, 1]$, and are a popular tool for modelling the joint behaviour of random variables. Their main feature is that they are able to model the dependence structure independently from the marginal models. The main theory of bivariate copulae is introduced in Section 3.1, and common estimation and inference procedures are discussed in Section 3.2. In Section 3.3 we introduce a method to build multidimensional models based on combining bivariate copulae and special graph models called vines. When building vines, there are many possibilities for construction. In Section 3.4, we introduce some methods for selecting and estimating a vine. Copula models are based on the assumption that data come from continuous distributions and therefore do not contemplate the presence of ties in the data set. In reality though, ties are always present. In Section 3.6 we discuss how the presence of ties influences parameter estimation and present possible solutions to improve the inference. Section 3.7 is a brief introduction to time series modelling in relation to copulae.

3.1 Definition and Basic Properties

The popularity of copulae in statistical modelling is due to the theorem introduced in Sklar (1959), which separates modelling of a multivariate distribution into two steps; the joint behaviour of the random variables and their *univariate* marginal distributions. This is achieved by letting the univariate distributions be joined by a d-dimensional copula C. In this section we only discuss the bivariate case. From a mathematical point of view a bivariate copula is defined as following:

Definition 3.1.1. *Copula: A 2-dimensional copula is a function* $C(u, v)$ *on* $[0, 1]^2 \to [0, 1]$ *that satisfies*

(i) $C(u_2, v_2) + C(u_1, v_1) - C(u_1, v_2) - C(u_2, v_1) \geq 0$, *for* $u_1 \leq u_2, v_1 \leq v_2$ *in* $[0, 1]^2$

(ii) $C(0, v) = C(u, 0) = 0$, *for all* $u, v \in [0, 1]$

(iii) $C(u,1) = u$ *and* $C(1,v) = v$*, for all* $u, v \in [0,1]$.

The popularity of copulae in statistical modelling is due to the following theorem introduced in Sklar (1959):

Theorem 1. *Let* H *be a bivariate distribution function with marginals* F *and* G*. Then there exists a copula* C *such that for all* x, y *in* $[-\infty, \infty]$*,*

$$H(x,y) = C(F(x), G(y)) = C(P[X \leq x], P[Y \leq y]). \tag{3.1}$$

If F *and* G *are continuous, then* C *is unique; otherwise* C *is uniquely determined on* $\mathrm{Ran}F \times \mathrm{Ran}G$*, where* $\mathrm{Ran}F$ *denotes the range. Conversely, if* C *is a copula and* F *and* G *are distribution functions, then the function* H *defined by* (3.1) *is a joint distribution function with univariate margins* F *and* G*.*

In essence, Sklar's theorem states that it is possible to model the dependence structure between the random variables X and Y in two separate steps. The dependence captured by the copula is independent of the marginals, and can thus be estimated separately. Notice also that the margins F and G can be distributions from different families.

Copulae are differentiable for almost all $u, v \in [0,1]$ so the density function of the copula can be obtained by:

$$c(u,v) = \frac{\partial^2 C(u,v)}{\partial u \partial v}. \tag{3.2}$$

Now the joint density for x and y is found by applying the chain rule:

$$h(x,y) = \frac{\partial^2 C(F(x), G(y))}{\partial u \partial v} = c(F(x), G(y)) f(x) g(y). \tag{3.3}$$

The conditional copula distribution can be obtained by:

$$C_{v|u}(v|u) = \frac{\partial C(u,v)}{\partial u}, \tag{3.4}$$

which can also be used to find the conditional joint distribution by applying the chain rule as before.

When analyzing the dependence between two random variables, there are two limiting cases: (i) the variables are independent, (ii) the variables are a function of each other. Both cases can be represented by copulae.

For the first case, the independence copula is given by $\Pi(u,v) = uv$, and the following theorem states a correspondence between X and Y being stochastically independent, and the Π copula:

Theorem 2. *(Nelsen, 2006, Theorem 2.4.2) Let* X *and* Y *be continuous random variables. Then* X *and* Y *are independent if and only if* $C_{XY} = \prod$*.*

At the other extreme, if X and Y are deterministic monotonic functions of each other, it follows that their dependence structure must be represented by one of the following copulae:

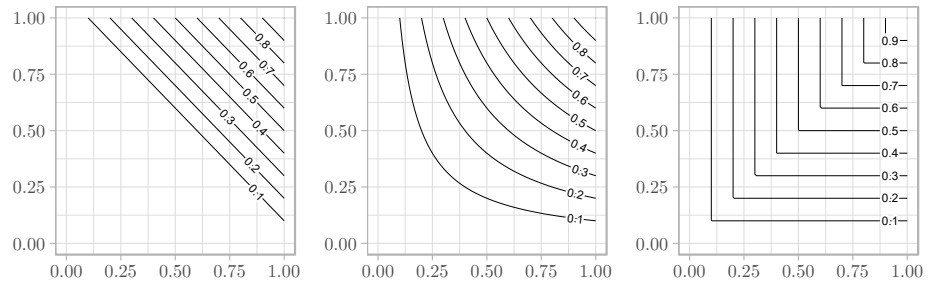
$$W(u,v) = \max(u + v - 1, 0) \tag{3.5a}$$

$$M(u,v) = \min(u,v) \tag{3.5b}$$

The copula W captures the decreasing behaviour, while M captures the increasing behaviour. These upper and lower bounds for the copula are known as *Fréchet-Hoeffding bounds*:

$$W(u,v) \leq C(u,v) \leq M(u,v). \tag{3.6}$$

Figure 3.1 shows contours of the copulae of these bounds, and the independence copula.



(a) W **(b)** Π **(c)** M

Figure 3.1: Contour plots of the limiting case copulae W, Π and M, respectively.

The most common families of copulae are the elliptical and Archimedean copulae. The elliptical copulae are generalizations of multivariate Gaussian and student-t distributions. Table 3.1 shows some one parameter Archimedean copulae and the Gaussian copula. For

Copula	$C(u,v)$	Parameter Range
AMH	$\frac{uv}{1-\theta(1-u)(1-v)}$	$\theta \in [-1,1)$
Clayton	$\left[\max\left\{u^{-\theta}+v^{-\theta}-1;0\right\}\right]^{-1/\theta}$	$\theta \in [-1,\infty) \setminus \{0\}$
Frank	$-\frac{1}{\theta}\log\left[1+\frac{(\exp(-\theta u)-1)(\exp(-\theta v)-1)}{\exp(-\theta)-1}\right]$	$\theta \in \mathbb{R} \setminus \{0\}$
Gumbel	$\exp\left[-\left((-\log(u))^{\theta}+(-\log(v))^{\theta}\right)^{1/\theta}\right]$	$\theta \in [1,\infty)$
Joe	$1-\left[(1-u)^{\theta}+(1-v)^{\theta}-(1-u)^{\theta}(1-v)^{\theta}\right]^{1/\theta}$	$\theta \in [1,\infty)$
Gaussian	$\frac{1}{\sqrt{1-\theta^2}}\exp\left\{-\frac{\theta^2\left(x_1^2+x_2^2\right)-2\theta x_1 x_2}{2(1-\theta^2)}\right\}$	$\theta \in (-1,1)$

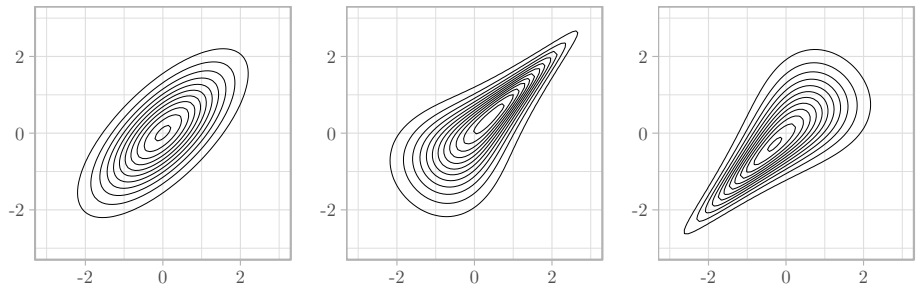
Table 3.1: Table of some Archimedian copula and the Gaussian copula. θ is the copula parameter, and for the Gaussian copula $x_1 = \Phi^{-1}(u)$ and $x_2 = \Phi^{-1}(v)$, where $\Phi^{-1}(\cdot)$ denotes the inverse standard normal distribution (Aas et al., 2009).

the Archimedean and Gaussian copulae it is possible to define an explicit link between the copula parameter θ and the strength of dependence measured by Kendall's τ = $g(\theta)$. This relationship can be used to compute the copula parameter as $\theta = g^{-1}(\tau)$. Table 3.2 shows the functions $\tau = g(\theta)$. For the copulae considered, g is a monotonically increasing function of θ, hence, an increase in θ indicates stronger dependence. See the books Nelsen (2006); Joe (1997) for details regarding the specific copulae. Figure 3.2 shows the densities of the Gaussian (elliptical), Joe and Clayton copula (Archimedean). These illustrate

Copula	$g(\theta)$	Dependence Range
AMH	$1 - \frac{2}{3\theta} - \frac{2(1-\theta)^2}{3\theta^2}\log(1-\theta)$	$\tau \in \left[\frac{(5-8\log 2)}{3}, \frac{1}{3}\right]$
Clayton	$\frac{\theta}{\theta+2}$	$\tau \in [-1, 1)$
Frank	$1 - \frac{4}{\theta}\left[1 - D_1(\theta)\right]$	$\tau \in [-1, 1)$
Gumbel	$\frac{\theta-1}{\theta}$	$\tau \in [0, 1)$
Joe	$1 - 4\sum_{k=1}^{\infty} 1/[k(\theta k+2)\{\theta(k-1)+2\}]^*$	$\tau \in [0, 1)$
Gaussian	$\frac{2}{\pi}\arcsin\theta^{**}$	$\tau \in (-1, 1)$

Table 3.2: Table of the relationship between the copula parameter θ and Kendall's τ, $\tau = g(\theta)$. $D_k(\theta) = k/\theta^k \int_0^\theta (t/\theta)/(e^t - 1)\,\mathrm{d}t$ is the Debye function, defined for any positive integer k, see (Nelsen, 2006). *(Hofert et al., 2012),**(Cramér, 1946).

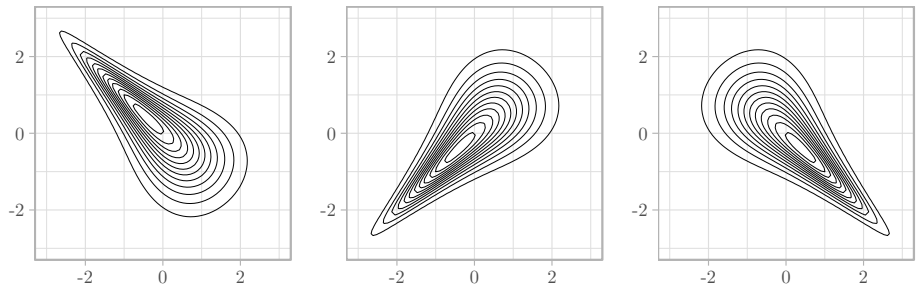
some of the different dependence structures copulae are able to model. Notice that the Gaussian copula shows a symmetric dependence, whereas the Joe and Clayton copulae have increasing depence towards the upper and lower tails (tail dependence), respectively.



(a) Gaussian **(b)** Joe **(c)** Clayton

Figure 3.2: Contour plots of the Gaussian, Joe and Clayton copula densities for Kendall's $\tau = 0.5$ with standard normal margins.

The different family of copulae have different ranges of dependence which they are able to model, i.e. Joe's copula can by default only model positive dependence. However, copulae can be rotated, which gives access to negative dependence structure of such copulas. From Sklar's theorem (1), we see that if we let $u = F(x), v = G(y)$, where $u, v \in [0, 1]$, we can rotate the data, or flip the axis, by letting $1 - u = F(x), v = G(y)$. Here the first axis has been flipped, and will from here be referred to as 90° (counterclockwise) rotation. Following this notion, $1 - u = F(x), 1 - v = G(y)$ is a 180° rotation, and $u = F(x), 1 - v = G(y)$ a 270° rotation. Figure 3.3 illustrates this rotation for the Joe copula.



(a) 90° **(b)** 180° **(c)** 270°

Figure 3.3: Rotated Joe copulae with rotations $90°, 180°$ and $270°$ with Kendall's $\tau = -0.5, 0.5$ and -0.5. The margins are standard normal.

3.2 Estimation of Bivariate Copulae

In this section we introduce the steps for estimating and selecting a bivariate copula. In 3.2.5, we also introduce a goodness-of-fit test to evaluate whether the fit of the selected copula is good.

3.2.1 Test of Independence

Before estimating a copula, we want to check if there is an underlying dependence to be modelled. Here we show a computationally simple test, which is based on the asymptotic normality of the sample version of Kendall's τ. The test is given by

$$\tau_n = \frac{2}{n(n-1)} \sum_{i<j} \text{sgn}(x_i - x_j)\text{sgn}(y_i - y_j), \tag{3.7}$$

where n denotes the number of observations. Under the null hypothesis of independence, τ_n will have mean 0, and the sample variance given by $2(2n+5)/(9n(n-1))$, which allows for a test of independence by the asymptotic normality of the test statistic

$$T = \sqrt{\frac{9n(n-1)}{2(2n+5)}} \times |\tau_n|. \tag{3.8}$$

The p-value is calculated as

$$\text{p-value} = 2 \times (1 - \Phi(T)), \tag{3.9}$$

where Φ is the standard normal distribution (Genest & Favre, 2007). This calculation is computationally inexpensive and can be used to avoid the computations involved in estimating copulae. There are more advanced test available, i.e. in Kojadinovic & Yan (2010), but these are more computationally costly, and this test is assumed to be sufficient for constructing larger models in Section 3.4. It should be noted that Kendall's τ is unable to

measure non-monotonic dependence, so the variables may still be dependent even if the indicates the opposite. This is illustrated by letting $\mathbf{X}$ be a vector of equally spaced observations in the interval $[-5, 5]$, and letting $\mathbf{Y} = \mathbf{X}^2$. The variables are clearly dependent, while having Kendall's $\tau = 0$.

3.2.2 Estimation

If pairs of stochastic variables are found to be dependent, a copula can be estimated. In the literature there are different methods for estimating the copula, such as inversion of the relationship $\theta = g^{-1}(\tau)$ introduced in Section 3.1, see Schölzel & Friederichs (2008); Genest & Favre (2007). However, estimation by maximum pseudo-likelihood is arguably the most common, and is required to apply the interval censoring in Section 3.6.1, so other techniques will not be discussed in detail. The marginal distributions are generally unknown, so to avoid misspecification, these are estimated by the ranks of the observations, which can described as follows: let $\boldsymbol{X}$ be a vector of size n, the Rank(X_i)= R_i, for $i = 1, \ldots, n$ is the number of entries in X smaller than X_i plus one. So max(X) would have rank n, while the smallest would have rank 1. This completely separates the modelling of the marginal distributions and the underlying joint copula. For an absolutely continuous copula C_θ, with density c_θ, the pseudo-loglikelihood function is then given by

$$l(\theta) = \sum_{i=1}^{n} \ln \left\{ c_\theta \left(\frac{R_{i1}}{n+1}, \ldots, \frac{R_{id}}{n+1} \right) \right\}, \tag{3.10}$$

where R_{ij} denotes the rank of X_{ij} among $\{X_{1j}, \ldots, X_{nj}\}$ where $1 \leq i \leq n$. This function is then maximized to obtain the parameter estimates. This normalized rank transformation of the observations is some times referred to as the *pseudo-observation*. As discussed in Genest & Favre (2007), rank based copula estimation retains the most statistical information (Oakes, 1982). Notice that this is in essence the log-likelihood function with the empirical distribution function as the marginal distribution, normalized by $n + 1$ instead of n to avoid problems at the boundary (Kojadinovic & Yan, 2010).

3.2.3 Selection

While it is possible to visually inspect the data, and fit the appropriate copulae according to the suspected dependence, it is usually more efficient to fit all available copulae and choose the the best fit in an automated procedure. Selection based on the lowest Akaike Information Criterion (AIC) was found to be most accurate in a large simulation study performed by Brechmann (2010). The AIC is given by

$$\mathrm{AIC} = 2k - 2\ln(\hat{L}), \tag{3.11}$$

where $\hat{L}$ is the maximum likelihood estimate of the model, and k is the number of parameters in the model (Akaike, 1974). It should be noted that while selection by AIC is mostly sufficient, there exist more advanced tools for copula selection that may perform better for a given task, see Grønneberg & Hjort (2014) and Ko et al. (2019). The AIC is easily applicable with the interval censored estimation in Section 3.6.1, as it does not require additional modification.

3.2.4 Sampling

Once we have fitted and selected a copula, the next step is to assess the fit. This can be done by a comparison with pseudo-random samples generated from the copula. Generating pseudo-random samples is also essential for the goodness-of-fit test introduced in Section 3.2.5. For the bivariate case, the pseudo-random sampling can be performed by the the inverse probability integral transform of Devroye (1986). Let w_1 and w_2 be two independent pseudo-random samples, we can use the inverse conditional copula distribution, from eq. (3.4), to generate random samples x_1 and x_2 from a copula $C(u, v)$:

$$x_1 = w_1 \text{ and } x_2 = C_{2|1}^{-1}(w_2|x_1) = \frac{\partial C^{-1}(w_1, w_2)}{\partial u}, \tag{3.12}$$

and x_1, x_2 are now samples from the copula C with uniform marginals.

3.2.5 Goodness-of-Fit tests

A goodness-of-fit test for copulae is a more formal apporoach for assessing whether the estimated copula is in fact the underlying copula. In this section we introduce one goodness-of-fit test, which can be applied to all types of copulae. The test is based on a parametric bootstrapping scheme. In Section 3.6.2, we show modifications to the test which increases its power in the presence of ties. There are other tests available, such as the White test (Huang & Prokhorov, 2014; White, 1982), however, this requires computation of the Hessian matrix, and is not generalized to account for ties.

For a fitted copula C_θ, a goodness of fit can be based on a comparison with the empirical copula C_n given by

$$C_n(u, v) = \frac{1}{n} \sum_{i=1}^{n} 1 \left(\frac{R_i}{n+1} \leq u, \frac{S_i}{n+1} \leq v \right), \tag{3.13}$$

where $1(A)$ denotes the indicator function for a set (A). The empirical copula is a rank based asymptotic estimator of the underlying copula (Deheuvels, 1979, 1981). Under the null hypothesis that $H_0 : C \in \{C_\theta\}$, that is, that the unique underlying copula C is in fact in the family of the fitted copula C_θ, the test can be based on the empirical process

$$\mathbb{C}_n(\mathbf{u}) = \sqrt{n}\{C_n(\mathbf{u}) - C_\theta(\mathbf{u})\}, \quad \mathbf{u} \in [0, 1]^d. \tag{3.14}$$

If we let the parameter estimate $\hat{\theta}$ be estimated by ranks, the following statistic was found to give the best results by Monte Carlo experiments performed by Berg (2009) and Genest et al. (2009).

$$S_n = \int_{[0,1]^d} \mathbb{C}_n(\mathbf{u})^2 \, \mathrm{d}C_n(\mathbf{u}) = \sum_{i=1}^{n} \left\{ C_n(\hat{\mathbf{U}}_i) - C_\theta(\hat{\mathbf{U}}_i) \right\}^2, \tag{3.15}$$

where $\hat{\mathbf{U}}$ denotes pseudo-observations. The underlying distribution can then be approximated by a parametric bootstrap procedure in order to obtain a p-value, see Genest &

Rémillard (2008). The same can also be done by a multiplier central limit theorem sampling procedure, see Kojadinovic et al. (2011), which for large samples, where the parametric bootstrap procedure is too computationally in-feasible, is significantly faster.

While this test is rather computationally demanding, it is also more generally applicable than the White test as the power is less dependent on the number of observations and does not set restrictions to the differentiability of the copula. One issue is that p-values are inaccurate when the number of parametric boostrapping samples are fewer than 10 times the number of observations, which can be very computationally infeasible (Genest et al., 2009).

3.3 Multivariate Copulae

While there are a large number of bivariate copula models whose properties have been explored in detail Nelsen (2006); Joe (1997); Genest et al. (2006); Li et al. (2016), the expansion to multivariate copula models is far from straight forward. There have been several attempts at such constructions (Bandeen-Roche & Liang, 1996; Joe, 1997; McNeil, 2008), but these are theoretically demanding compared to the bivariate case and not very flexible when dimensions are large. One way to construct complex, multivariate models using bivariate copulae as building blocks was proposed by Aas et al. (2009). We first give some intuition on how a larger multivariate distribution can be decomposed into smaller bivariate distributions, before giving a more formal definition. Let $h(x_1, \dots, x_d)$ be a d-dimensional multivariate density function. This can be factorized as

$$h(x_1, \dots, x_d) = f_1(x_1) \cdot f(x_2|x_1) \cdot f(x_3|x_1, x_2) \cdots f(x_d|x_1, \dots, x_{d-1}), \tag{3.16}$$

which is a unique decomposition until we relabel the variables (Aas et al., 2009). For $d = 4$ we have:

$$h(x_1, x_2, x_3, x_4) = f_1(x_1) f_{2|1}(x_2|x_1) f_{3|1,2}(x_3|x_1, x_2) f_{4|1,2,3}(x_4|x_1, x_2, x_3), \tag{3.17}$$

which is essentially a product of four univariate distributions. By Sklar's Theorem (1), copulae separate the joint and marginal behaviour. Following Eq. (3.3), each conditional density in (3.16) can be written as the product of a copula and a marginal distribution:

$$\begin{aligned} f_{2|1}(x_2|x_1) &= \frac{c_{1,2}\{F_1(x_1), F_2(x_2)\} f_1(x_1) f_2(x_2)}{f_1(x_1)} \\ &= c_{1,2}\{F_1(x_1), F_2(x_2)\} f_2(x_2). \end{aligned} \tag{3.18}$$

This allows us to write the joint density as

$$\begin{aligned} h_{1,2,3,4} = & f_1(x_1) f_2(x_2) f_3(x_3) f_4(x_4) \\ & c_{1,2}\{F_1(x_1), F_2(x_2)\} c_{2,3}\{F_2(x_2), F_3(x_3)\} c_{3,4}\{F_3(x_3), F_4(x_4)\} \\ & c_{1,3|2}\{F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2)\} c_{2,4|3}\{F_{2|3}(x_2|x_3), F_{4|3}(x_4|x_3)\} \\ & c_{1,4|2,3}\{F_{1|2,3}(x_1|x_2, x_3), F_{4|2,3}(x_4|x_2, x_3)\}, \end{aligned} \tag{3.19}$$

which is the product of univariate marginal distributions and bivariate copulae. Such decompositions are commonly referred to as vine copulae, since the distributions are illustrated as trees. Each stochastic variable is represented by a node, which are joined by an edge representing a bivariate copula. The density (3.19) is illustrated as trees in Figure 3.4.

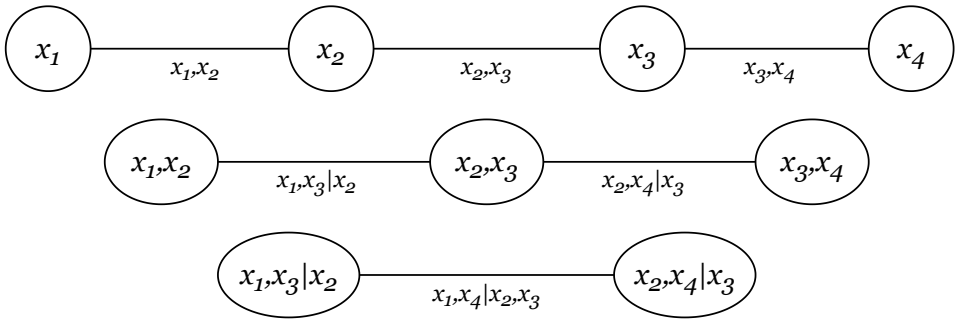


Figure 3.4: Example of a four dimensional vine copula density illustrated as trees. This is commonly referred to as a four dimensional D-vine, which is a special case of an R-vine.

The parametrization in (3.19) is not unique as we can relabel the variables. This means that the number of decompositions for a vector of d variables is of order $d!$. Therefore we need a strategy for vine construction, and the most general is arguably the regular vine (R-vine). The R-vine distributions were introduced in Bedford & Cooke (2001, 2002) as a generalization to Markov trees, and were later given more detailed attention in Kurowicka & Cooke (2006). In Dißmann et al. (2012), the authors proposed automated methods for construction and sampling, which even when simplified, yield better results than other automated alternatives (Brechmann et al., 2012).

An R-vine $\mathcal{V}$ on d variables, as defined in (Kurowicka & Cooke, 2006, Definition 4.4), is built by $T_1, \dots, T_{d-1}$ trees with nodes N_i and edged E_i for $i = 1, \dots, d-1$ that satisfy the following requirements:

Definition 3.3.1. ***R-vine (Kurowicka & Cooke, 2006, Chapter 4.4)***
$\mathcal{V}$ is an R-vine on d elements if

1. *T_1 is a spanning tree with nodes $N_1 = \{1, \dots, d\}$ and a set of edges denoted E_1.*
2. *For $i = 2, \dots, d-1$, T_i is a spanning tree with nodes $N_i = E_{i-1}$ and edge set E_i.*
3. *For $i = 2, \dots, d-1$ and edges $\{a, b\} \in E_i$, it must hold that the two edges $\{a, b\}$ share a common node in tree T_{i-1} (proximity condition)*

A spanning tree is a graph that join all nodes in the tree with minimum possible edges. So for a tree on d variables, the spanning tree will have $d-1$ edges. The R-vine $\mathcal{V}$ structure consisting of the node set $\mathcal{N} \coloneqq \{N_1, \dots, N_{d-1}\}$ with the edge set $\mathcal{E} \coloneqq \{E_1, \dots, E_{d-1}\}$ is the framework for building a larger statistical model. Each individual edge $e \in E_i$ of tree $i \in 1, \dots d-1$ can be specified as $e = j(e), k(e)|D(e)$ and joins two nodes, where $j(e)$ and $k(e)$ denote the *conditioned* stochastic variables conditioned on the set $D(e)$, denoted

the *conditioning set*. As an example, consider the final edge $x_1, x_4|x_2, x_3$ in Figure 3.4. Here $j(e) = 1$, $k(e) = 4$, which are the free variables, conditioned on $D(e) = 2, 3$. This edge joins the nodes $x_1, x_3|x_2$ and $x_2, x_4|x_3$. The notation for the R-vine is general, but in essence express that the variables $j(e), k(e)$ conditioned on the set $D(e)$ form a constraint set that is exclusive to each edge.

Vines can be used to select a decomposition of the multivariate distribution. We build the joint distribution by letting the nodes be associated with stochastic variables, and the edges with bivariate copula densities. From Theorem 4.2 of Kurowicka & Cooke (2006) there is a proof that the resulting multivariate regular vine density of a random variable $\boldsymbol{X}$ is uniquely determined by and given by

$$h(x_1, \ldots, x_d) = \left[\prod_{k=1}^{d} f_k(x_k)\right] \times \left[\prod_{i=1}^{d-1} \prod_{e \in E_i} c_{j(e),k(e)|D(e)}(F(x_{j(e)}|\boldsymbol{x}_{D(e)}), F(x_{k(e)}|\boldsymbol{x}_{D(e)}))\right], \quad (3.20)$$

where $\boldsymbol{x}_{D(e)}$ denotes the subvector of $\boldsymbol{x} = (x_1, \ldots, x_d)^\top$ determined by the indices in $D(e)$. The density is the product of the marginal densities, and the product of all copula densities in the R-vine tree.

Notice that for the trees $T_2, \ldots, T_{d-1}$, the copulae take conditional distributions as arguments, since for these trees, the conditioning sets $D(e)$ are not empty. These conditional distributions $F_{j(e)|D(e)}, F_{k(e)|D(e)}$ depend on the copulae in the previous trees, and are thus defined recursively. If we let $\mathbf{v}$ be a vector, where v_j is the element j in the vector and $\mathbf{v}_{-j}$ is the vector *without* element j. The conditional distributions $F(x_i|\mathbf{v})$ can be derived as

$$F(x_i|\mathbf{v}) = \frac{\partial C_{x_i,v_j|\mathbf{v}_{-j}}\left\{F(x_i|\mathbf{v_{-j}}), F(v_j|\mathbf{v_{-j}})\right\}}{\partial F(v_j|\mathbf{v}_{-j})}, \quad (3.21)$$

where $C_{ij|\mathbf{k}}$ is a bivariate copula, see Joe (1996) for the proof. For the first tree, when v is univariate, the conditional distributions are given by

$$F(x_i|v_j) = \frac{\partial C_{x_i,v_j}\left\{F(x_i), F(v_j)\right\}}{\partial F(v_j)}. \quad (3.22)$$

We demonstrate the recursion on the copula $C_{13|2}$ shown in Figure 3.4. The copula density is given by

$$c_{13|2}\Big(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2)\Big) = c_{13|2}\left\{\frac{C_{12}(F_1(x_1), F_2(x_2))}{\partial F_2(x_2)}, \frac{C_{23}(F_2(x_2), F_3(x_3))}{\partial F_2(x_2)}\right\} \quad (3.23)$$

Now we have written a distribution of size d as a product of univariate marginal distributions and bivariate copulae.

The example in Figure 3.4 is usually referred to as a drawable vine (D-vine), which is a special case of the R-vine. Each node in a D-vine has two edges at max. In four

dimensions, we can also construct the vine as in Figure 3.5. This is usually referred to as a canonical vine (C-vine), which is also a special case of the R-vine. The first tree of a C-vine has one canonical node with an edge to all other nodes.

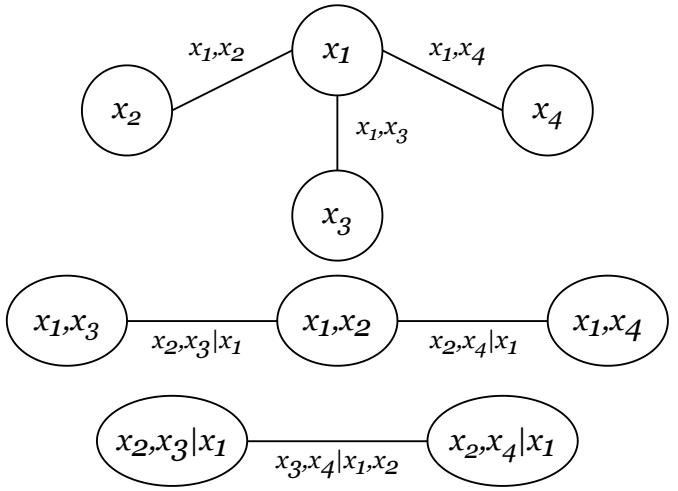


Figure 3.5: Example of a four dimensional vine illustrated as trees. This is commonly referred to as a C-vine which is a special case of the R-vine.

In four dimensions, however, the possible vines are still fairly limited. Therefore we show a possible R-vine in seven dimensions in Figure 3.6. More details regarding R-vines can be found in Dißmann et al. (2012), and more intuition on pair-copula decomposition of a large multivariate distribution in Aas et al. (2009).

3.4 Vine Selection and Estimation

In this section we introduce the steps of building a vine copula model using the automated "top-down" approach of Dißmann et al. (2012). The process generally consists of two steps iterated until all trees are built. Firstly, we construct a tree by maximizing over a set of edge weights. Secondly, a copula is selected to each edge following the steps for bivariate copulae, as described in Section 3.2. These steps are then iterated to the tree is built. Commnly, tree selection is done by strength of correlation, and we start by computing Kendall's τ for all parameter pairs. The next steps can be described as follows:

1. Construct the first spanning tree T_1 by maximizing the set of edge weights

$$\max \sum_e w_{j(e),k(e)|D(e)}, \tag{3.24}$$

for some edge weight $w_{j(e),k(e)|D(e)}$. Commonly strength of dependence, $w = |\hat{\tau}|$, where $\hat{\tau}$ denotes the empirical Kendall's τ. The edge weights can be maximized according to (Cormen et al., 2009, p. 631) from Kruskal (1956).

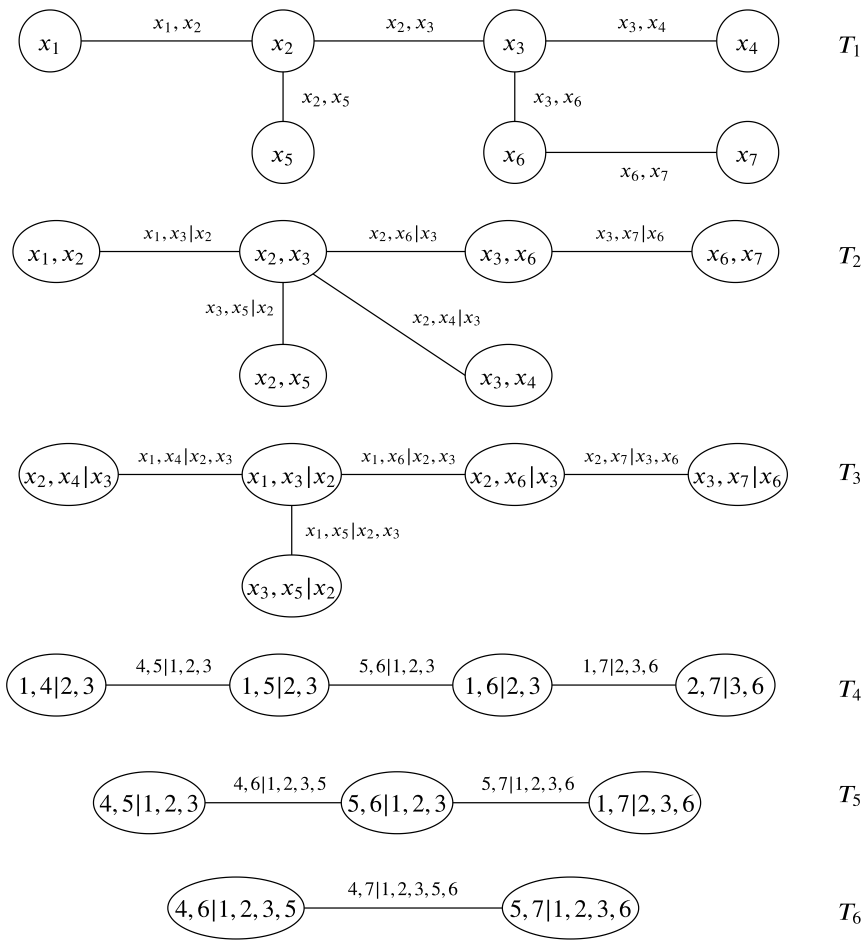


Figure 3.6: Example of a possible R-vine in seven dimensions (Dißmann et al., 2012, Figure 1). From T_4, the variables are shown by their index, i.e. $j(e), k(e)|D(e)$. Each edge e corresponds to a copula for the variables $(x_{j(e)}, x_{k(e)}|x_{D(e)})$, and the nodes correspond to stochastic variables.

2. For each edge e in the spanning tree T_1, find a bivariate copula according to some selection criterion. Commonly first by a test of independence, and then by AIC if the pair is dependent. We estimate each copula by maximum pseudo-likelihood.

3. Find all possible edges in the next tree and compute the edge weights. For selection by Kendall's τ, the edge weights would be the estimated conditional correlations

$$w_{j(e),k(e)|D(e)} = |\hat{\tau}|_{j(e),k(e)|D(e)} = \left| \hat{\tau} \left(F_{j(e)|D(e)}, F_{k(e)|D(e)} \right) \right|. \tag{3.25}$$

This evaluation is recursive, since it depends on the previous trees, see Eq. (3.21). The first tree only requires the marginal distributions, which we estimate by pseudo-observations.

4. Continue with these steps until all trees constructed.

The underlying assumption in the "top-down" approach is that by first selecting the first level tree, followed by the second, third etc., most of the dependence is "captured" in the early trees, thus maximizing according to Kendall's τ seems like a natural choice. Depending on the application of the vine copula, other weights might be more appropriate, for instance tail dependence in financial applications (Dißmann et al., 2012; Brechmann, 2010).

To avoid unnecessarily large and complex models, the data for each edge is tested for independence by applying the test from Section 3.2.1. If the variables are found to be independent, thus best described by the independence copula Π, this reduces the model complexity and in turn the computational costs. While this test is simple, we assume that it is sufficient for construction of larger models. Reduction of computational complexity can also be achieved by substituting copulae in later trees with the Gaussian copula in a process referred to as truncation, see Brechmann et al. (2012). This is most relevant for data with large dimensionality, and is not discussed here.

Notice also that in this procedure, we only need to compute the pseudo-observations *once*, which is when we find copulae to each edge in the first tree. For the sequential trees, copulae are selected based on a nested evaluation of these data, that is in step 3. These evaluations can also be stored, and used for sequential trees. We will refer to evaluations of the conditional distribution as conditional data. As an example:

$$X_{1|2} = F_{1|2}(X_1|X_2) = \frac{\partial C_{1,2}\left\{\texttt{pobs}(X_1), \texttt{pobs}(X_2)\right\}}{\partial \texttt{pobs}(X_2)}, \tag{3.26}$$

where `pobs` denotes computing the pseudo-observations. When we adapt interval censoring to vines, step 3 requires more attention. This will be discussed in detail when interval censoring is adapted to vines in Section 3.6.3.

Each selected copula can be tested for goodness-of-fit with the test from Section 3.2.5. This a *sequential* vine copula estimation, which as fast since we only perform bivariate estimation. Algorithms are described in more detail in Dißmann et al. (2012).

3.4.1 Joint Estimation

Up until now, the sequential estimation approach has been presented. That is, we have obtained pairwise maximum-likelihood estimates, and not the maximum likelihood estimates for the full model. A joint model estimation can be performed by maximum likelihood estimation of the full R-vine density (3.20). The log-likelihood can be expressed by

$$l(\theta) = \sum_{i=1}^{n} \left\{ \sum_{i=1}^{d-1} \sum_{e \in E_i} \log c_{j(e),k(e)|D(e)}(F(x_{j(e)}|\boldsymbol{x}_{D(e)}), F(x_{k(e)}|\boldsymbol{x}_{D(e)})) \right\}, \tag{3.27}$$

where $\boldsymbol{x}_{D(e)}$ denotes the subvector of $\boldsymbol{x} = (x_1, \ldots, x_d)^\top$ determined by the indices in $D(e)$, as described in Section 3.3. We see that log-likelihood is the sum of the log-likelihood for each bivariate copula in the tree. For a vine of dimension three, the log

likelihood can be written as

$$l(\theta) = \sum_{i=1}^{n} \Big\{ \log c_{13}\Big(F_1(X_{i1}), F_3(X_{i3})\Big) + \log c_{23}\Big(F_1(X_{i2}), F_3(X_{i3})\Big) + \log c_{12|3}\Big(F_{1|3}(X_{i1}|X_{i3}), F_{2|3}(X_{i2}|X_{i3})\Big)\Big\}.$$

The performance of the estimation techniques were explored in Haff (2012), and the sequential approach is mostly sufficient. It loses some asymptotic efficiency for increasing dependence, in particular in the later trees. The number of model parameters grows exponentially with increasing data dimensionality, so for certain applications, performing the sequential estimation may be the only viable option.

3.5 Vine Sampling

Similar to bivariate copula, we want to generate samples to assess the model fit, and for applications such as modelling compound extreme events, as in Bevacqua et al. (2017). Sampling schemes are also necessary to conduct the simulation study in Chapter 4.4.4. Samples from a vine can be generated similarily to the bivariate case, from Section 3.2.4, by "inverse transform sampling" (Devroye, 1986). If we let $w_1, \dots, w_d$ be uniform independent random samples on $[0, 1]$, the vine samples $x_1, \dots, x_d$ can be computed as

$$\begin{aligned} x_1 &= w_1 \\ x_2 &= F_{2|1}^{-1}(w_2|x_1) \\ &\vdots \\ x_n &= F_{d|1,\dots,d-1}^{-1}(w_d|x_1, \dots, x_{d-1}) \end{aligned} \tag{3.28}$$

In three dimensions this results in

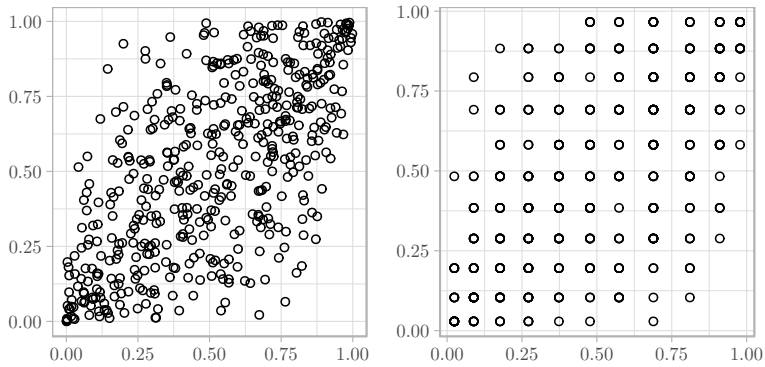
$$\begin{aligned} x_1 &= w_1 \\ x_2 &= F_{2|1}^{-1}(w_2|w_1) \\ x_3 &= F_{3|1,2}^{-1}(w_3|F_{2|1}^{-1}(w_2|w_1)). \end{aligned}$$

A detailed algorithm for regular vines can be found in Dißmann et al. (2012).

3.6 Ties

The copulae discussed here have continuous marginal distributions, and a consequence is that the probability of ties in the observations is equal to *zero*. As in Hofert et al. (2018), we say that t a d-dimensional data set of size n $\boldsymbol{X_1}, \dots, \boldsymbol{X_n}$ contains ties if at least one component of an observation contains ties. Even though processes are continuous, however, ties may still occur as a result of in imprecisions in the measurement techniques or rounding. We estimate copula by pseudo-observations (`pobs`), as described in Section

3.2, which are essentially normalized ranks. Ranks are not unique in the presence of ties, but there are some simple methods commonly suggested for managing tied ranks. The first is calculating the average rank of the smaller and larger ranks, and assigning the average value to the tied ranks. That is, for a tied random vector $\boldsymbol{x} = \{1, 3, 3, 3, 5, 6, 8, 8, 9\}$ the average ranks are $\{1, 3, 3, 3, 5, 6, 7.5, 7.5, 9\}$. Pseudo-observations computed from average ranks are denoted ($\texttt{pobs}_{\text{avg}}$). The issue with this method is, however, that the data is still tied, thus having locally stronger dependence than in reality. We illustrate this by collecting 500 samples from a Gaussian copula with Kendall's $\tau = 0.5$, and rounding both margins to the first decimal. The original and rounded samples are shown in Figure 3.7. The empirical Kendall's τ (3.7) is $\hat{\tau} = 0.464$ for the original data, and $\hat{\tau} = 0.495$ for the average ranks of the rounded samples. The estimated dependence from average ranks is *larger* than for the original samples.

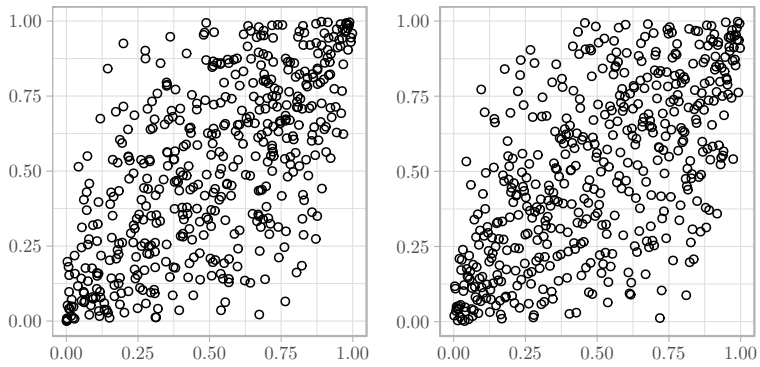


(a) Original samples, $\hat{\tau} = 0.464$ **(b)** Average ranks, $\hat{\tau} = 0.495$

Figure 3.7: Illustration of bias introduced by ties on the Gaussian copula. In (b), both margins are rounded to the first decimal, and this increases the estimated dependence.

Another method is to assign ranks randomly, which will always result in untied data. That is, for a tied random vector $\boldsymbol{x} = \{1, 3, 3, 3, 5, 6, 8, 8, 9\}$ the random ranks can be $\{1, 3, 2, 4, 5, 6, 7, 8, 9\}$. However, here we introduce *independence* to the data. For the same rounded samples as before, this process is shown in Figure 3.8. Now the data resembles the original samples more, but the estimated dependence $\hat{\tau} = 0.448$ has decreased. One can carry out estimation with randomization a number of times, and average over the results, but for data of high dimensionality, the number of possible random outcomes increases greatly, and cannot be considered as a reliable method (Hofert et al., 2018).

Since estimation of bivariate copula heavily relies on pseudo-observations which are standardized ranks, the presence of ties can introduce bias in parameter estimation. An alternative to using average or random ranks is introduced in Li et al. (2016), and will be introduced in the next section.



(a) Original samples, $\hat{\tau} = 0.464$ **(b)** Random ranks, $\hat{\tau} = 0.448$

Figure 3.8: Illustration of bias introduced by randombly breaking ties on the Gaussian copula. In (b), both margins are rounded to the first decimal, followed computing pseudo-observations by randomly breaking ties. This decreases the estimated dependence.

3.6.1 Interval Censoring

In the cases where the observations contain ties, we can think of ranks as "censored" by their upper and lower limits. That is, we know which interval the rank belongs to, but not necessarily which rank it is. The approach introduced in Li et al. (2016) is called interval censoring, and is in essence a modification to the likelihood function such that each observation is assigned a likelihood based on the interval of its censoring limits. That is, the parameters are estimated by maximum pseudo-likelihood calculated from both the upper limit (max) and the lower limit (min) of the ranks. For the random vector $\boldsymbol{x} = \{1, 3, 3, 3, 5, 6, 8, 8, 9\}$, the minimum ranks are $\{1, 2, 2, 2,\ 5, 6, 7, 7, 9\}$ and the maximum ranks are $\{1, 4, 4, 4, 5, 6, 8, 8, 9\}$. Let $(\overline{U}, \overline{V})$ denote the upper censoring limits computed from maximum rank pseudo-observations ($\texttt{pobs}_{\max}$), and let $(\underline{U}, \underline{V})$ be the lower censoring limits computed from minimum rank pseudo-observations ($\texttt{pobs}_{\min}$). Note that these limits are equal if the observations are not tied. Each pseudo-observation's (U_i, V_i) contribution to the pseudo-likelihood function can be divided into four cases:

1. If $\underline{U_i} < \overline{U_i}$ and $\underline{V_i} < \overline{V_i}$, both margins are tied, then

$$L_i(\theta) = C_\theta(\overline{U_i}, \overline{V_i}) - C_\theta(\overline{U_i}, \underline{V_i}) - C_\theta(\underline{U_i}, \overline{V_i}) + C_\theta(\underline{U_i}, \underline{V_i}). \tag{3.29}$$

2. If $\underline{U_i} < \overline{U_i}$ and $\overline{V_i} = \underline{V_i} = V_i$, that is only tied in the first margin, then

$$L_i(\theta) = \frac{\partial C_\theta(u, v)}{\partial v}\Big|_{u=\overline{U_i}, v=V_i} - \frac{\partial C_\theta(u, v)}{\partial v}\Big|_{u=\underline{U_i}, v=V_i}. \tag{3.30}$$

3. If $\overline{U_i} = \underline{U_i} = U_i$ and $\underline{V_i} < \overline{V_i}$, that is only tied in the second margin, then

$$L_i(\theta) = \frac{\partial C_\theta(u, v)}{\partial u}\Big|_{u=U_i, v=\overline{V_i}} - \frac{\partial C_\theta(u, v)}{\partial u}\Big|_{u=U_i, v=\underline{V_i}}. \tag{3.31}$$

4. If $\overline{U_i} = \underline{U_i} = U_i$ and $\overline{V_i} = \underline{V_i} = V_i$, that is tied in neither margin, then

$$L_i(\theta) = c_\theta(U_i, V_i). \tag{3.32}$$

which gives the resulting pseudo-likelihood function

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} \log L_i(\theta), \tag{3.33}$$

which can be optimized using a standard maximum likelihood approach. In essence, the observations are now assigned a likelihood based on an interval. Note that in the case of no ties, i.e. the fourth case, the likelihood is the standard likelihood function.

3.6.2 Bootstrapping With Ties

The goodness-of-fit test and p-values in Section 3.2.5 are based on bootstrap replicates from the fitted model. Informally, this scheme can be regarded as a *comparison* of a test statistic from the fitted model, with test statistics from models fitted to samples from this. When the data is untied, the estimated marginal distributions, i.e. the ordered pseudo-observations, will *always* be the same. Moreover, sampled values will always be untied, and have the same estimated marginal distribution. Note that the source of variation in the bootstrapping scheme, comes from how the samples are paired in the joint distribution. In the presence of ties, however, the untied samples will have a different estimated marginal distribution than in the original data, and the scheme gives a less direct comparison. That is, the estimates from these samples will not be interval censored, and the test statistic (3.15) will not be calculated with any tied samples. In Li et al. (2016), the authors therefore suggest a procedure which preserves the empirical distribution function of the original data. If we let $\tilde{F}_n$ and $\tilde{G}_n$ denote the empirical distributions of the original pseudo observations, that is $\tilde{F}_n(u) = \sum_{i=1}^{n} \mathbf{1}(U_i \leq u)/n$, and similar for $\tilde{G}$, we can introduce ties by computing the corresponding quantile functions to the bootstrapped samples. That is, if we let $U_i^{(b)}$ and $V_i^{(b)}$ be bootstrapped pseudo-observations generated from the fitted copula, we transform the observations as follows:

$$\left(U_i^{(b)}, V_i^{(b)}\right) \leftarrow \left(\tilde{F}_n^{-1}(U_i^{(b)}), \tilde{G}_n^{-1}(V_i^{(b)})\right), \tag{3.34}$$

for $i = 1, \ldots, n$, where $\tilde{F}_n^{-1}(y) = \inf\{u : \tilde{F}_n(u) \geq y\}$, i.e. the inverse of the original empirical distribution. Now the bootstrapped pseudo-observations have the same marginal empirical distribution functions as the original data (Bücher & Kojadinovic, 2015), and the models are estimated by interval censoring.

3.6.3 Interval Censored Vines

In this section we will attempt to apply interval censoring to vines. Interval censoring is limited by the fact that one has to compute cross partial derivatives, which is challenging for dimensions higher than 2. However, the advantage of vine constructions is that we may apply theory from bivariate copuale on larger multivariate distributions. Partial derivatives

have to be calculated to find the conditional distributions (3.21) regardless, so the excess work is limited. For each bivariate copula in the vine, we need upper and lower limits for the tied intervals to adjust the likelihood (3.33) in four cases. For bivariate copulae, this is done by estimating the marginal distributions by pseudo-observations from max and min ranks. As mentioned in Section 3.4, this step is only required once for vine copulae, which in practice is when we construct the first tree. Later trees are, however, estimated from nested evaluations of these pseudo-observations, by the conditional distribution formula (3.21), which do *not* maintain the upper and lower limits from the first tree. As an example, we demonstrate on the three dimensional vine in Figure 3.9. If we want to estimate an

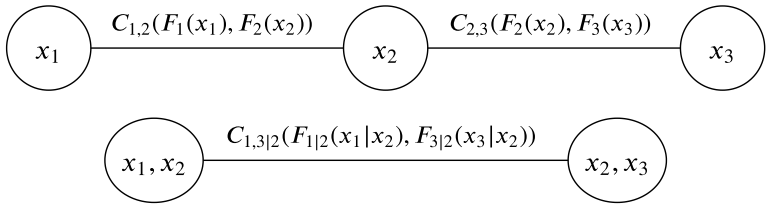


Figure 3.9: Example of a three dimensional vine copula.

interval censored copula for $C_{1,3|2}$, we first compute

$$\begin{aligned} x_{1|2} &= \frac{C_{1,2}\{F_1(x_1), F_2(x_2)\}}{\partial F_2(x_2)} \\ x_{3|2} &= \frac{C_{2,3}\{F_2(x_2), F_3(x_3)\}}{\partial F_2(x_2)}, \end{aligned} \tag{3.35}$$

in which the copulae $C_{1,2}, C_{2,3}$ can be estimated using standard bivariate interval censoring, as in Section 3.6.1, since these belong to the first tree. This allows us to estimate $C_{1,3|2}(x_{1|2}, x_{3|2})$. One could assume that that the upper and lower censoring limits of $C_{1,3|2}$ could simply be given by letting the marginal distributions F_1, F_2, F_3 be estimated by both maximum and minimum ranks. That is, if we let $C_{1,2}(u, v)$ be a Joe copula with parameter $\theta = 1.5$, and let $\texttt{pobs}_{\max}(X_{1,2}) = (0.81, 0.91)$, $\texttt{pobs}_{\min}(X_{1,2}) = (0.80, 0.90)$ be some tied data, this approach would give the following censoring limits for $x_{1|2}$:

$$\begin{aligned} \overline{U}_{1|2} &= \frac{C_{1,2}\{\texttt{pobs}_{\max}(X_1), \texttt{pobs}_{\max}(X_2)\}}{\partial \texttt{pobs}_{\max}(X_2)} \\ &= \frac{\partial C_{1,2}(0.81, 0.91)}{\partial v} = 0.5785 \\ \underline{U}_{1|2} &= \frac{C_{1,2}\{\texttt{pobs}_{\min}(X_1), \texttt{pobs}_{\min}(X_2)\}}{\partial \texttt{pobs}_{\min}(X_2)} \\ &= \frac{\partial C_{1,2}(0.80, 0.90)}{\partial v} = 0.5867, \end{aligned} \tag{3.36}$$

and here the lower limit is larger than the upper. Note that the selection of the tied data here is to illustrate a case where the limits are not maintained. This does not always happen,

however, which can be seen if we let $\texttt{pobs}_{\max}(X_{1,2}) = (0.61, 0.71)$, $\texttt{pobs}_{\min}(X_{1,2}) = (0.60, 0.70)$, which gives the limits

$$\begin{aligned}
\overline{U}_{1|2} &= \frac{C_{1,2}\{\texttt{pobs}_{\max}(X_1), \texttt{pobs}_{\max}(X_2)\}}{\partial \texttt{pobs}_{\max}(X_2)} \\
&= \frac{\partial C_{1,2}(0.61, 0.71)}{\partial v} = 0.5717 \\
\underline{U}_{1|2} &= \frac{C_{1,2}\{\texttt{pobs}_{\min}(X_1), \texttt{pobs}_{\min}(X_2)\}}{\partial \texttt{pobs}_{\min}(X_2)} \\
&= \frac{\partial C_{1,2}(0.60, 0.70)}{\partial v} = 0.5670.
\end{aligned} \tag{3.37}$$

Here $\overline{U}_{1|2} > \underline{U}_{1|2}$, which is different from the previous case, and we see that adjustments have to be made.

Two methods are suggested for interval censoring vines, which will be illustrated on the copula $C_{13|2}$. In the first method, denoted *simple censoring*, we estimate the marginal distributions by computing pseudo-observations as usual, but we also estimate the upper and lower censoring limits of copulae in $\{T_2, \ldots, T_{d-1}\}$ by pseduo-observations. Now pseudo-observations are computed twice for copulae with non-empty conditioning sets $D(e)$. For $x_{1|2}$ this would give the censoring limits:

$$\begin{aligned}
\overline{U}_{1|2} &= \texttt{pobs}_{\max}\left\{\frac{C_{1,2}\{\texttt{pobs}(X_1), \texttt{pobs}(X_2)\}}{\partial \texttt{pobs}(X_2)}\right\} \\
\underline{U}_{1|2} &= \texttt{pobs}_{\min}\left\{\frac{C_{1,2}\{\texttt{pobs}(X_1), \texttt{pobs}(X_2)\}}{\partial \texttt{pobs}(X_2)}\right\}
\end{aligned} \tag{3.38}$$

and for $x_{3|2}$

$$\begin{aligned}
\overline{V}_{3|2} &= \texttt{pobs}_{\max}\left\{\frac{C_{2,3}\{\texttt{pobs}(X_2), \texttt{pobs}(X_3)\}}{\partial \texttt{pobs}(X_2)}\right\} \\
\underline{V}_{3|2} &= \texttt{pobs}_{\min}\left\{\frac{C_{2,3}\{\texttt{pobs}(X_2), \texttt{pobs}(X_3)\}}{\partial \texttt{pobs}(X_2)}\right\}
\end{aligned} \tag{3.39}$$

where pobs is computed from some tie preserving rank method, i.e average ranks. The idea behind this method stems from the fact that as long as ties are preserved, tie preserving pseudo-observations will be the same. That is, $\texttt{pobs}_{\max}(x) = \texttt{pobs}_{\max}(\texttt{pobs}_{\text{avg}}(x))$. Since ties are preserved when computing conditional distributions, we are still able to censor the intervals in bivariate estimation. The censored intervals $(\underline{U}_{1|2}, \overline{U}_{1|2}), (\underline{V}_{3|2}, \overline{V}_{3|2})$ can now be used as in the bivariate case.

In higher trees, the censoring limits are specified via the conditional distributions derived from Eq. (3.21), which are essentially nested evaluations of the marginal distributions. The censoring limits of *simple censoring* can more generally be given by a similar formula. Let $\mathbf{v}$ be a vector, where v_j is the element j in the vector and $\mathbf{v}_{-j}$ is the vector *without* element j. The conditional interval censoring limits for *simple censoring* are given

by

$$\overline{U}_{x_i|\mathbf{v}} = \texttt{pobs}_{\max}\left\{\frac{\partial C_{x_i,v_j|\mathbf{v}_{-j}}\left\{\hat{F}(x_i|\mathbf{v}_{-\mathbf{j}}),\hat{F}(v_j|\mathbf{v}_{-\mathbf{j}})\right\}}{\partial \hat{F}(v_j|\mathbf{v}_{-j})}\right\} \tag{3.40a}$$

$$\underline{U}_{x_i|\mathbf{v}} = \texttt{pobs}_{\min}\left\{\frac{\partial C_{x_i,v_j|\mathbf{v}_{-j}}\left\{\hat{F}(x_i|\mathbf{v}_{-\mathbf{j}}),\hat{F}(v_j|\mathbf{v}_{-\mathbf{j}})\right\}}{\partial \hat{F}(v_j|\mathbf{v}_{-j})}\right\}, \tag{3.40b}$$

where $\hat{F}$ denotes that the marginal distributions in the first tree are estimated by pseudo observations from average ranks

$$\hat{F}(x_i|v) = \frac{\partial C_{x_i,v}\left\{\texttt{pobs}_{\text{avg}}(x_i),\texttt{pobs}_{\text{avg}}(v)\right\}}{\partial \texttt{pobs}_{\text{avg}}(v)}. \tag{3.41}$$

However, estimation of the censoring limits by computing new pesudo-observations for each tree, could cause a "smoothing". To emphasize this, let $X_{1|2}$ = (0.01, 0.4, 0.4, 0.99) be some conditional data, which would give the maximum pseudo observation $\texttt{pobs}_{\max}(X_{1|2}) = (1,3,3,4)/5 = (0.2, 0.6, 0.6, 0.8)$ and the minimum $\texttt{pobs}_{\min}(X_{1|2}) = (1,2,2,4)/5 = (0.2, 0.4, 0.4, 0.8)$, which is quite far from the original conditional data. This is mainly an issue for small sample sizes.

It should also be noted that while this method requires a tie preserving method for computing ranks, i.e. max, min or average ranks, the methods are not equal in this regard. That is, while $\texttt{pobs}_{\max}(x)$ = $\texttt{pobs}_{\max}(\texttt{pobs}_{\text{avg}}(x))$ holds,

$$\begin{aligned}\overline{U}_{1|2} &= \texttt{pobs}_{\max}\left\{\frac{C_{1,2}\{\texttt{pobs}_{\text{avg}}(X_1),\texttt{pobs}_{\text{avg}}(X_2)\}}{\partial \texttt{pobs}_{\text{avg}}(X_2)}\right\} \\ &= \texttt{pobs}_{\max}\left\{\frac{C_{1,2}\{\texttt{pobs}_{\max}(X_1),\texttt{pobs}_{\max}(X_2)\}}{\partial \texttt{pobs}_{\max}(X_2)}\right\},\end{aligned}$$

does *not* always hold. Say we want to compute $x_{1|2}$, and let $C_{1,2}$ be a Clayton copula with parameter $\theta = 3$. If we have one one untied pseudo-observation $(U_1, V_1) = \{0.2, 0.2\}$ and one tied $(\hat{U}_2, \hat{V}_2) = \{0.4, 0.4\}$ (average), $(\overline{U}_2, \overline{V}_2) = \{0.4, 0.6\}$ (max), the untied conditional observation would be $x_{1|2}(U_1, V_1) = 0.399$, and the tied observations $x_{1|2}(\hat{U}_2, \hat{V}_2) = 0.414$ and $x_{1|2}(\overline{U}_2, \overline{V}_2) = 0.150$. The resulting rank of the observations would change depending on the method used for managing ties, and this will ultimately affect the estimation by (3.40). However, this may not occur on less severely tied data.

In the second method, denoted *full censoring*, we follow the same steps as in the example (3.36). The conditional data computed from max ranks is not always larger than the conditional data computed from min ranks, so we let the upper and lower limits be the

maximum and minimum of the two. For $x_{1|2}$ this would give

$$\begin{aligned}
\overline{U}_{1|2} = \max \Bigg\{ & \frac{C_{1,2}\{\mathtt{pobs}_{\max}(X_1), \mathtt{pobs}_{\max}(X_2)\}}{\partial \mathtt{pobs}_{\max}(X_2)}, \\
& \frac{C_{1,2}\{\mathtt{pobs}_{\min}(X_1), \mathtt{pobs}_{\min}(X_2)\}}{\partial \mathtt{pobs}_{\min}(X_2)} \Bigg\} \\
\underline{U}_{1|2} = \min \Bigg\{ & \frac{C_{1,2}\{\mathtt{pobs}_{\max}(X_1), \mathtt{pobs}_{\max}(X_2)\}}{\partial \mathtt{pobs}_{\max}(X_2)}, \\
& \frac{C_{1,2}\{\mathtt{pobs}_{\min}(X_1), \mathtt{pobs}_{\min}(X_2)\}}{\partial \mathtt{pobs}_{\min}(X_2)} \Bigg\}
\end{aligned} \tag{3.42}$$

and for $x_{3|2}$

$$\begin{aligned}
\overline{V}_{3|2} = \max \Bigg\{ & \frac{C_{2,3}\{\mathtt{pobs}_{\max}(X_2), \mathtt{pobs}_{\max}(X_3)\}}{\partial \mathtt{pobs}_{\max}(X_2)}, \\
& \frac{C_{2,3}\{\mathtt{pobs}_{\min}(X_2), \mathtt{pobs}_{\min}(X_3)\}}{\partial \mathtt{pobs}_{\min}(X_2)} \Bigg\} \\
\underline{V}_{3|2} = \min \Bigg\{ & \frac{C_{2,3}\{\mathtt{pobs}_{\max}(X_2), \mathtt{pobs}_{\max}(X_3)\}}{\partial \mathtt{pobs}_{\max}(X_2)}, \\
& \frac{C_{2,3}\{\mathtt{pobs}_{\min}(X_2), \mathtt{pobs}_{\min}(X_3)\}}{\partial \mathtt{pobs}_{\min}(X_2)} \Bigg\}
\end{aligned} \tag{3.43}$$

where max and min denote the maximum and minimum of each observation. Using similar notation as in (3.40), the conditional interval censoring limits for *full censoring* are given by:

$$\overline{U}_{x_i|\mathbf{v}} = \max \left\{ \frac{\partial C_{x_i,v_j|\mathbf{v}_{-j}}\{\overline{F}(x_i|\mathbf{v}_{-\mathbf{j}}), \overline{F}(v_j|\mathbf{v}_{-\mathbf{j}})\}}{\partial \overline{F}(v_j|\mathbf{v}_{-j})}, \frac{\partial C_{x_i,v_j|\mathbf{v}_{-j}}\left\{\underline{F}(x_i|\mathbf{v}_{-\mathbf{j}}), \underline{F}(v_j|\mathbf{v}_{-\mathbf{j}})\right\}}{\partial \underline{F}(v_j|\mathbf{v}_{-j})} \right\} \tag{3.44a}$$

$$\underline{U}_{x_i|\mathbf{v}} = \min \left\{ \frac{\partial C_{x_i,v_j|\mathbf{v}_{-j}}\{\overline{F}(x_i|\mathbf{v}_{-\mathbf{j}}), \overline{F}(v_j|\mathbf{v}_{-\mathbf{j}})\}}{\partial \overline{F}(v_j|\mathbf{v}_{-j})}, \frac{\partial C_{x_i,v_j|\mathbf{v}_{-j}}\left\{\underline{F}(x_i|\mathbf{v}_{-\mathbf{j}}), \underline{F}(v_j|\mathbf{v}_{-\mathbf{j}})\right\}}{\partial \underline{F}(v_j|\mathbf{v}_{-j})} \right\} \tag{3.44b}$$

where $\overline{F}$ and $\underline{F}$ denotes that the marginal distributions in the first tree are estimated from maximum and minimum rank pseudo-observations as in (3.41). Note that the max/min step is only performed in the copula estimation, and the resulting $\overline{U}, \overline{V}, \underline{U}, \underline{V}$ are not stored

for computing conditional data in later trees. Now the implications of the upper and lower limits are preserved in the whole tree, and the censoring is less reliant on the arbitrary choice of a ties managing method for generating conditional data. This method may, however, increase the estimation difficulty, since copula densities are evaluated closer to the boundaries, and numerical approximations of the gradient and hessian are less reliable.

3.7 Time Series Analysis

In order to apply the theory of copula, the observations have to independent identically distributed (iid) and the underlying distribution function has to be continuous. The assumption of iid random variables can fail in more ways, and in this section we look at such situation that are relevant to time series analysis. Weather events could be considered a stretch from a multivariate time series, and in this section we will look at stationarity, multiple serial independence and the ARIMA-model.

3.7.1 Stationarity

Informally we can say that stationarity in a time series is that consecutive observations do not follow a trend. More formally, let the observations $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ be a stretch from a multivariate time series $(\boldsymbol{X}_i)_{i\in\mathbb{Z}}$, it is said to be *stationary* if for any $k \in \mathbb{N}$ and $m \in \mathbb{Z}$, the vector $(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k)$ and $(\boldsymbol{X}_{1+m}, \ldots, \boldsymbol{X}_{k+m})$ have the same distribution (Hofert et al., 2018). This is referred to as *strong* stationarity in the literature.

One way to assess whether the time series is stationary is by applying a test. There exist many such tests, but in an extensive simulation study carried out in Bücher et al. (2019) not all were found to maintain their levels. Testing whether a times series is stationary in full generality is difficult, and as suggested in Hofert et al. (2018), an imperfect approach is to apply tests for *change point detection*, which are tests constructed from the following null hypothesis:

$$\begin{aligned}\mathcal{H}_0 : &\text{ There exists a distribution function } H \text{ such that} \\ &\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \text{ have a distribution function } H\end{aligned} \tag{3.45}$$

For an overview of the literature, see Csörgö & Horváth (1997) and Aue & Horváth (2013). The following test has a good sensitivity to departures from the null hypothesis, which is described as in Hofert et al. (2018). It can be derived from the empirical process

$$\begin{aligned}\mathbb{D}_n^H(t, \boldsymbol{x}) = \sqrt{n}\lambda_n(0,t)\lambda_n(t,1)\left(H_{1:\lfloor nt\rfloor}(\boldsymbol{x}) - H_{(\lfloor nt\rfloor+1):n}(\boldsymbol{x})\right), \\ (t, \boldsymbol{x}) \in [0,1]\times\mathbb{R}^d,\end{aligned} \tag{3.46}$$

where $\lambda_n(t,t') = (\lfloor nt'\rfloor - \lfloor nt\rfloor)/n$ and $\lfloor\cdot\rfloor$ denotes the floor function, and for any $1 \le k \le l \le n$, let

$$H_{k:l}(\boldsymbol{x}) = \frac{1}{l-k+1}\sum_{i=k}^{l}\mathbf{1}(\boldsymbol{X}_i \le \boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}, \tag{3.47}$$

be the empirical distribution function from the subsample $\boldsymbol{X}_k, \ldots, \boldsymbol{X}_l$ obtained from $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$. The suggested test statistic is then

$$S_n^H = \sup_{t \in [0,1]^d} \int_{\mathbb{R}^d} \left(\mathbb{D}_n^H(t, \boldsymbol{x}) \right)^2 \mathrm{d}H_{1:n}(\boldsymbol{x}) = \max_{1 \leq k \leq n-1} \frac{1}{n} \sum_{i=1}^{n} \left(\mathbb{D}_n^H(k/n, \boldsymbol{X}_i) \right)^2. \quad (3.48)$$

The p-value can then be approximated by a resampling procedure.

While this test has shown strength in detecting departures from the null hypothesis of an underlying multivariate distribution, it is mostly in regards to the marginal distributions, and not changes in the underlying copula (Holmes et al., 2013). One test particularly sensitive to such changes is the one suggested in Bücher et al. (2014). From Sklar's Theorem (1), we have that the distribution function H can be written as a function of a unique underlying copula C and the marginal distribution functions $F_1, \ldots, F_d$, and it follows that we can decompose the null hypothesis (3.45) into $\mathcal{H}_0 : \mathcal{H}_{0,m} \cap \mathcal{H}_{0,c}$, where

$$\begin{aligned} \mathcal{H}_{0,m} &: \text{ There exist } F_1, \ldots, F_d \text{ such that } \boldsymbol{X}_1, \ldots, \boldsymbol{X}_d \text{ have marginals } F_1, \ldots, F_d, \\ \mathcal{H}_{0,c} &: \text{ There exists a copula } C \text{ such that } \boldsymbol{X}_1, \ldots, \boldsymbol{X}_d \text{ have a copula } C. \end{aligned} \quad (3.49)$$

Now we want to construct a hypothesis test with the alternative hypothesis $\mathcal{H}_1 : \mathcal{H}_{0,m} \cap (\neg \mathcal{H}_{0,c})$, where $\neg$ denotes negation. That is a test that only checks for the existence of a copula, when the marginals are assumed to exist. The test, as described in Hofert et al. (2018), is similar to (3.48), and derived from the following empirical process

$$\mathbb{D}_n^C(t, \boldsymbol{u}) = \sqrt{n} \lambda_n(0, t) \lambda_n(t, 1) \left(C_{1:\lfloor nt \rfloor}(\boldsymbol{u}) - C_{(\lfloor nt \rfloor + 1):n}(\boldsymbol{u}) \right), \quad (t, \boldsymbol{u}) \in [0,1]^{d+1}, \quad (3.50)$$

where for any $1 \leq k \leq l \leq n$,

$$C_{k:l}(\boldsymbol{u}) = \frac{1}{l - k + 1} \sum_{i=k}^{l} \mathbf{1}(\boldsymbol{U}_i^{k:l} \leq \boldsymbol{u}), \quad \boldsymbol{u} \in [0,1]^d, \quad (3.51)$$

is the empirical copula of the subsample $\boldsymbol{X}_k, \ldots, \boldsymbol{X}_l$ obtained from $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$. Note that it is the convention to let $C_{k:l} = 0$ if $l < k$. The sample of pseudo-observations $\boldsymbol{U}_k^{k:l}, \ldots, \boldsymbol{U}_l^{k:l}$ is given by

$$\boldsymbol{U}_i^{k:l} = (F_{k:l,1}(X_{i1}), \ldots, F_{k:l,d}(X_{id})) \frac{l - k + 1}{l - k + 2}, \quad i \in \{k, \ldots, l\}, \quad (3.52)$$

where $F_{k:l,j}$ is the empirical distribution function of $X_{kj}, \ldots, X_{lj}$. The suggested test statistic is

$$S_n^C = \sup_{t \in [0,1]} \int_{[0,1]^d} \left(\mathbb{D}_n^C(t, \boldsymbol{u}) \right)^2 \mathrm{d}C_{1:n}(\boldsymbol{u}) = \max_{1 \leq k \leq n-1} \frac{1}{n} \sum_{i=1}^{n} \left(\mathbb{D}_n^C(k/n, \boldsymbol{U}_i^{1:n}) \right)^2. \quad (3.53)$$

The test is based on resampling, and more details can be found in Bücher et al. (2014).

3.7.2 Multiple Serial Independence

If the time series is found to be stationary, the next step is to evaluate whether it is serially independent, i.e. if the observations have significant correlations at certain time lags. Given a sequence of stationary continuous random variables $X_1, \ldots, X_{n'}$, and some embedding dimension $p > 1$, the first step consists of forming $n = n' - p + 1$ p-dimensional vectors

$$\boldsymbol{Y}_i = (X_i, \ldots, X_{i+p-1}), \quad i \in \{1, \ldots, n\}, \tag{3.54}$$

for $i \in \{1, \ldots, n\}$. Singular serial independence can now be measured by the following statistic

$$I_n^s = \int_{[0,1]^p} n \left\{ C_n^s(\boldsymbol{u}) - \prod_{k=1}^{p} u_k \right\}^2 \mathrm{d}\boldsymbol{u}, \tag{3.55}$$

where C_n^s is the serial empirical copula computed from $\boldsymbol{Y}_i$ for $i = 1, \ldots, n$. The embedding dimension p also determines the maximum lag considered in the serial copula. This test is based on the empirical process $\sqrt{n}(C_n - \Pi)$, which converges weakly to the tight centered Gaussian process (Kojadinovic & Yan, 2010). For details regarding the convergence, see Stute (1984). In essence, this test is a comparison between the empirical copula and the independence copula. In Genest & Rémillard (2004) the authors proposed the use of the Möbius transform $M_{A,n}^s$, see Rota (1964), to decompose the process into $2^d - d - 1$ sub-processes $\sqrt{n}\mathcal{M}_A(C_n)$, for all non-empty subsets $A \subseteq \{1, \ldots, d\}$, with test statistics

$$M_{A,n}^s = \int_{[0,1]^d} n \left(\mathcal{M}_{A,n}(C_n^s)(\boldsymbol{u}) \right)^2 \mathrm{d}\boldsymbol{u}, \tag{3.56}$$

and under the null hypothesis of independence, the test statistics are asymptotically mutually independent.

This approach can then be generalized to the multivariate case using the permutation principle, see Kojadinovic & Yan (2011). Following the decomposition, the individual test statistics can then be combined in to a global test statistic following combination rules such as Tippett (1931) or Fisher (1932). Note that the test of multiple serial independence relies on resampling to approximate p-values, so it can be quite computationally demanding. One solution is to reduce the cardinality m of the subsets considered when deriving the test statistics (3.56).

3.7.3 The ARIMA-Model

In cases where the data has significant non-stationarity or serial dependence, this can be transformed into a stationary serially independent time series by applying the *autoregressive integrated moving average* (ARIMA) model. The discussion of this model is based on the work by Brockwell & Davis (1987) and Shumway & Stoffer (2017).

Before giving the definition of the ARIMA process, we first define the ARMA process, for which we need to define the concepts autocovariance and white noise:

Definition 3.7.1. *(Brockwell & Davis, 1987, Definition 1.3.1)(The Autocovariance function). If $(\boldsymbol{X}_i)_{i\in T}$ is a process such that $\mathrm{Var}(\boldsymbol{X}_i) < \infty$ for each $i \in T$, then the autocovariance function $\gamma_X(\cdot,\cdot)$ of (X_i) is defined by*

$$\gamma_X(r,s) = \mathrm{Cov}(\boldsymbol{X}_r, \boldsymbol{X}_s) = E\left[(\boldsymbol{X}_r - E[\boldsymbol{X}_r])(\boldsymbol{X}_s - E[\boldsymbol{X}_s])\right], \quad r,s \in T. \tag{3.57}$$

For a stationary process, this can be written

$$\gamma_X(h) \equiv \gamma_X(h,0), \tag{3.58}$$

for some lag $h = r - s$. White noise can then be defined as

Definition 3.7.2. *(Brockwell & Davis, 1987, Definition 3.1.1) The process $\{\boldsymbol{Z}_t\}$ is said to be white noise with mean 0 and variance σ^2, written*

$$\{\boldsymbol{Z}_t\} \sim \mathrm{WN}(0,\sigma^2), \tag{3.59}$$

if and only if $\{\boldsymbol{Z}_t\}$ has zero mean and covariance function given by

$$\gamma(h) = \begin{cases} \sigma^2 & \text{if } h = 0 \\ 0 & \text{if } h \neq 0 \end{cases} \tag{3.60}$$

The next step is now to define the ARMA process:

Definition 3.7.3. *(Brockwell & Davis, 1987, Definition 3.1.2) The ARMA(p,q) process: The process $\{\boldsymbol{X}_t, t \in \mathbb{Z}\}$ is said to be an ARMA(p,q) process if $\{\boldsymbol{X}_t\}$ is stationary and if for every t,*

$$\boldsymbol{X}_t - \phi_1 \boldsymbol{X}_{t-1} - \cdots - \phi_p \boldsymbol{X}_{t-p} = \boldsymbol{Z}_t + \theta_1 \boldsymbol{Z}_{t-1} + \cdots + \theta_q \boldsymbol{Z}_{t-q}, \tag{3.61}$$

where $\{\boldsymbol{Z}_t\} \sim \mathrm{WN}(0,\sigma^2)$. We say that $\{\boldsymbol{X}_t\}$ is an ARMA(p,q) process with mean μ if $\{\boldsymbol{X}_t - \mu\}$ is an ARMA(p,q) process.

The equations (3.61) *can be written symbolically in more compact form*

$$\phi(B)\boldsymbol{X}_t = \theta(B)\boldsymbol{Z}_t, \quad t \in \mathbb{Z}, \tag{3.62}$$

where ϕ and θ are the p^{th} and q^{th} degree polynomials

$$\phi(z) = 1 - \phi_1 z - \cdots - \phi_p z^p \tag{3.63}$$

and

$$\theta(z) = 1 + \theta_1 z + \cdots + \theta_q z^q \tag{3.64}$$

and B is the backward shift operator defined by

$$B^j \boldsymbol{X}_t = \boldsymbol{X}_{t-j}, \qquad j \in \mathbb{Z}. \tag{3.65}$$

We now arrive at the ARIMA-process which generalizes the ARMA-process by introducing differencing:

Definition 3.7.4. *(Shumway & Stoffer, 2017, Definition 3.11) A process $(\boldsymbol{X}_i)_{i\in\mathbb{Z}}$ is said to be ARIMA(p, d, q)if*

$$\nabla^d \boldsymbol{X}_i = (1-B)^d \boldsymbol{X}_i \tag{3.66}$$

is ARMA(p, q), where ∇ denotes the differencing operator, i.e. $\nabla \boldsymbol{X}_i = \boldsymbol{X}_i - \boldsymbol{X}_{i-1}$.

For more details see Shumway & Stoffer (2017).

For time series with a complex, periodic annual trend, the data can be transformed by an ARIMA-model with fourier terms as external regression terms:

$$\boldsymbol{Y}_i = a + \sum_{k=1}^{K} \left[\alpha_k \sin(2\pi k i/m) + \beta_k \cos(2\pi k i/m)\right] + N_i, \tag{3.67}$$

where N_i is the ARIMA-process and m is the length of the period. The parameter K, and the ARIMA-model, can then be selected by minimal AIC (3.11). A similar method is used in Livera et al. (2011), but in a state space approach.

3.7.4 The Ljung-Box Test

To assess whether the fitted model is good, that is the resiuduals are iid, one can apply the Ljung-Box test (LJUNG & BOX, 1978), which considers the joint behaviour of the autocorrelations for different lags. If we let $\hat{\rho}_e$ denote the empirical autocorrelations of the residuals $(\hat{\boldsymbol{Z}})_{i\in\mathbb{Z}}$, under the null hypothesis of white noise, the Ljung-Box test statistic is given by

$$Q(\hat{\rho}) = n(n+2)\sum_{h=1}^{H} \frac{\hat{\rho}_e^2(h)}{n-h}, \tag{3.68}$$

for some maximum lag H, where the empirical residual autocorrelations $\hat{\rho}_e$ are given by

$$\hat{\rho}_e(h) = \frac{\sum_{t=h+1}^{n} \hat{\boldsymbol{Z}}_i \hat{\boldsymbol{Z}}_{i-h}}{\sum_{t=1}^{n} \hat{\boldsymbol{Z}}^2}. \tag{3.69}$$

The test statistic Q follows a χ^2_{H-p}-distribution, where p denotes the number of parameters in the model, typically $p+q$.

Chapter 4

Simulation Study

In this chapter, we conduct a simulation study on interval censored estimation. In Section 4.3 the performance is shown for bivariate estimation, in a study similar to Li et al. (2016), where the authors generated ties in copula samples by rounding the first margin to the first decimal place. In the hydrological data introduced in Chapter 2, ties are not only present in the first margin. Therefore we conduct these experiments with ties in both margins and different levels of severity. Furthermore, we use the sample sizes $n \in \{500, 1000, 5000\}$, as opposed to $n \in \{100, 200, 400\}$, to get a better representation of the samples sizes in our data. As in Li et al. (2016), we also conduct experiments were only the lower tails are rounded. In hydrological data, rounded lower tails are commonly present, hence, these experiments will be of specific interest. In Section 4.4 the performance is shown when the method is extended to vines. We proposed two methods for constructing censored vines, in Section 3.6.3, so the goal is to evaluate the performance of each method. The experiment is restricted to vine estimation for a given structure, and not the steps involved in vine selection. It is more difficult to construct simulations on selection *and* estimation of vines, so similar to Brechmann (2010), we perform a simulation study for bivariate copula selection under interval censoring. Selection by AIC was found to be simple and effective strategy, and the study verifies this result. Details can be found in Appendix A.1. For the estimation studies, we only show the main results in this section, the remaining are left in Appendix A.

4.1 A Note on the Implementation

The implementation was done in R (R Core Team, 2018), and while there are existing libraries for copula modelling, these do not support interval censoring. Ties are generally handled before applying any of the included functions, and since interval censoring requires adjustment of the likelihood function, most functions were implemented from scratch. Some parts are borrowed from the R-packages `copula` (Hofert et al., 2017; Yan, 2007; Kojadinovic & Yan, 2010; Hofert & Mächler, 2011) and `VineCopula` (Schepsmeier et al., 2018), specifically some function expression, such as copula densities, cop-

ula expressions and the partial derivatives, sampling procedures and some plotting and presentation functions. The package `igraph` (Csardi & Nepusz, 2006) was used to manage spanning trees, and the following are some packages used for purposes related to the implementation and presentation: `tidyverse` (Wickham, 2017) with `magrittr` (Bache & Wickham, 2014) and `lubridate` (Grolemund & Wickham, 2011) attached, `ggthemes` (Arnold, 2018).

The implementation includes methods for estimation, selection, R-vine construction and goodness-of-fit testing under interval censoring. All are implemented using maximum pseudo-likelihood for estimation by applying the base functions `optim` and `optimize` directly, that is by numerical approximations of the gradient and hessian. The copulae implemented were the AMH, Clayton, Frank, Gumbel, Joe, BB1, BB6, BB7, BB8, Tawn Type I and II, the Gaussian and Student-t copulae, see Nelsen (2006); Joe (1997). It should be noted that the family of Elliptical copulae, the Gaussian and Student t-copulae were time-consuming to estimate under interval censoring. In particular the student-t copula, as estimation times were typically around $60 - 70$ seconds per thousand observations in data with a considerable amount of ties. The difficulties were more apparent for strong correlations. Goodness-of-fit testing by parametric bootstrapping is most reliable when calculating ten times more bootstraps than observations (Genest et al., 2006), which can typically amount to around 13000 bootstraps, so for this reason the student-t copula is typically not included in the analysis. The Tawn type I and II copulae are also typically not included, since from testing, the global maximum likelihood was typically not *always* achieved. The AMH copula is not implemented in the package `VineCopula`, and thus typically not included in the analysis.

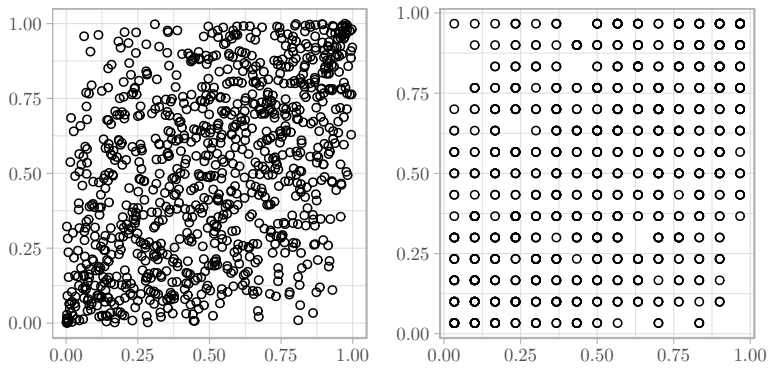
It should also be noted that the optimization difficulty increases for interval censored estimation, and is more prone to failure. Results that are clearly a consequence of optimization issues have been removed in this section. The code used in this chapter is a modified version of the full implementation. The full implementation is demonstrated in Chapter 5.3.

4.2 Experiment Design

In this section we describe the general structure of the experiments, starting by showing the two methods we use to induce ties into sampled data. The first method will be referred to as binning, in which the interval $[0, 1]$ is divided into b equally sized bins, and the values within each bin are assigned a common value, i.e. the the middle of the bin. Note that pseudo-observations are computed before estimation, such that the resulting values are given by the rank of each bin. Figure 4.1 illustrates this process for a Gaussian copula, where both margins are binned into $b = 15$ bins.

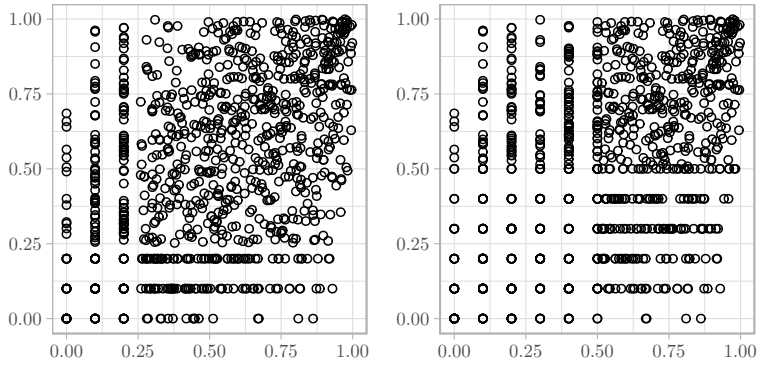
The second method consists of rounding a given percentage of the lower tails. Here, we round the percent λ of smallest samples to either the first or second decimal. This is to emulate rounding error in lower tails for different levels of severity. Figure 4.2 shows ties generated from this procedure for $\lambda \in \{0.25, 0.5\}$.

While the data at hand is typically tied in both margins, the amounts of ties might be different. We therefore conduct each experiment both symmetrically and asymmetrically. For the binning experiments, each margin is equally binned in the symmetrical case, while



(a) Non-tied samples **(b)** Tied samples

Figure 4.1: An illustration of binned ties generated in the Gaussian copula. In (**b**), both margins are binned with $b = 15$.

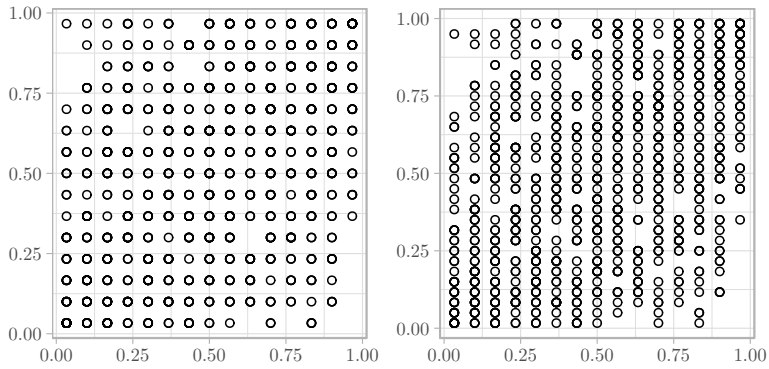


(a) $\lambda = 0.25$ **(b)** $\lambda = 0.5$

Figure 4.2: An illustration of lower tail ties generated in the Gaussian copula.

the second margin is typically binned into twice as many bins in the asymmetrical experiments. This is illustrated in Figure 4.3, and note that the second margin of Figure 4.3b is binned with $b = 30$, as opposed to $b = 15$. In the lower tail rounding experiments, the lower tails are rounded to one decimal in the symmetrical case, while in the asymmetric experiments, the lower tails of the first margin are typically rounded to one decimal, and the second to two decimals.

In each experiment, interval censored estimation is compared with estimation by average and random ranks. Each estimation by random ranks is computed as the mean estimate over 100 randomizations. All estimation methods are executed by maximum pseudo-likelihood (3.10). For the binning experiments, we measure performance for increasing sample size $n \in \{500, 1000, 5000\}$ and let the strength of correlation be specified by Kendall's τ indexed by $\tau \in \{0.1, 0.2, \ldots, 0.9\}$. In the tail rounding experiments, we use



(a) $b_1, b_2 = 15$ **(b)** $b_1 = 15, b_2 = 30$

Figure 4.3: An illustration of binned symmetrical and asymmetrical ties generated in the Gaussian copula.

the same sample sizes $n \in \{500, 1000, 5000\}$, strength of correlation $\tau \in \{0.25, 0.75\}$ and severity of tail ties $\lambda \in \{0.1, 0.2, \ldots, 0.5\}$. For these simulations, we restrict the copulae to the one parameter families introduced in Section 3.1, and compute the copula parameters θ from the given correlations by applying the links in Table 3.2. All experiments are conducted over $R = 1200$ repetitions. In some cases, estimation fails, so by computing 1200 repetitions, we are likely to get at least 1000 valid results for all methods. The results are shown as in Li et al. (2016) by boxplots for the error

$$\text{Error} = \theta - \hat{\theta},$$

which shows the estimation bias, and in some cases by the root mean square error

$$\text{RMSE} = \sqrt{\frac{1}{R}\sum_{i=1}^{R}\left(\theta - \hat{\theta}_i\right)^2},$$

which summarizes the estimation consistency. Here θ and $\hat{\theta}$ denote the true and estimated copula parameter, respectively. As mentioned, some of the results are left in Appendix A.

4.3 Bivariate Models

In this section we conduct two sets of experiments. The first involves inducing ties in the Gaussian copula by binning, whereas the second induces ties in the lower tails of the Joe and Clayton copulae. The dependence in Joe's copula grows stronger towards the upper tails, while dependence is stronger in the lower tails for Clayton. By rounding the lower tails, we censor the key features of the Clayton copula, but not the Joe copula.

Figure 4.4 and Figure 4.5 show the distribution of the estimation error $\theta - \hat{\theta}$ as a function of the correlation strength τ. The three panels refer to the three different sample sizes.

As mentioned in Section 3.1, the strength of dependence τ and the copula parameter θ have a monotonically increasing relationship, thus, an increase in θ would indicate stronger dependence. Figure 4.4 refers to the experiments where ties are generated by symmetric binning of a Gaussian copula. Here we see that with increasing strength of correlation average and random ranks tend to introduce bias in the estimation. In particular random ranks tend underestimate the correlation. Censoring appears to be unbiased for all τ, and this is most apparent when correlations are strong. Figure 4.5 refers to asymmetrically binned Gaussian copula and shows similar results.

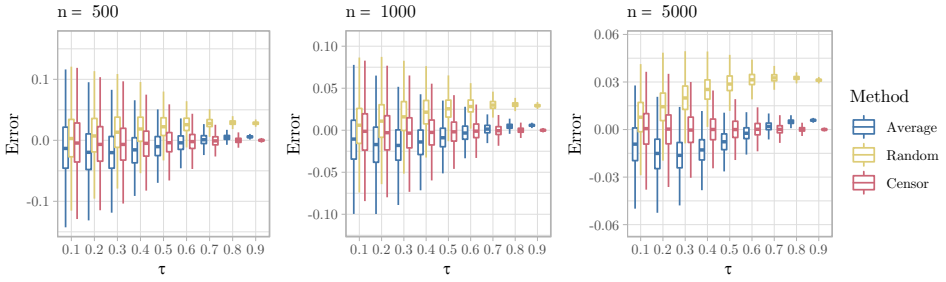


Figure 4.4: Boxplot of the estimation error $\theta - \hat{\theta}$ in the symmetrically binned Gaussian copula. Each margin is tied in $b = 15$ bins, for a given Kendall's τ and different sample sizes n.

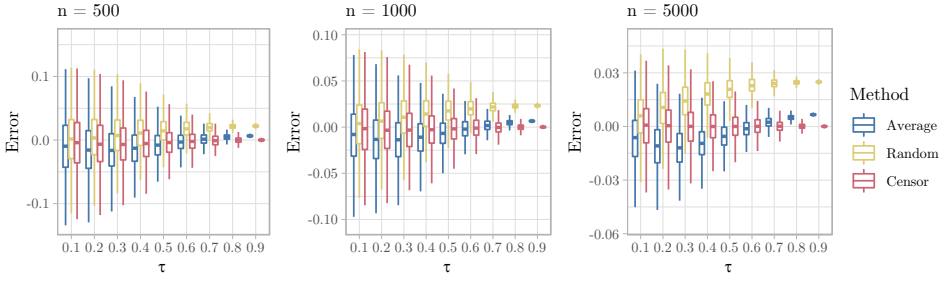


Figure 4.5: Boxplot of the estimation error $\theta - \hat{\theta}$ in the asymmetrically binned Gaussian copula. The first margin is tied in $b_1 = 15$ bins and the second in $b_2 = 30$ bins, for a given Kendall's τ and different sample sizes n.

Tail dependence is an important feature of certain copulae which distinguish these from other, and this is the emphasis of the second experiment. By rounding the lower tails, the key features of Joe's copula will not be censored, whereas the features for Clayton's copula will be. The error distribution and RMSE for these experiments on the Joe copula are shown in Figures 4.6 and 4.7. Note that now the first axis indicates severity of ties in the lower tails, and not strength of dependence. Both random and average ranks underestimate the correlation when the severity of ties λ increases, but not by much. For larger sample size ($n = 5000$) the RMSE stay fairly constant with respect to the increased severity of ties for the interval censored estimation. The differences are more clear as the sample size

increases, which is an indication of the inherent bias of the methods.

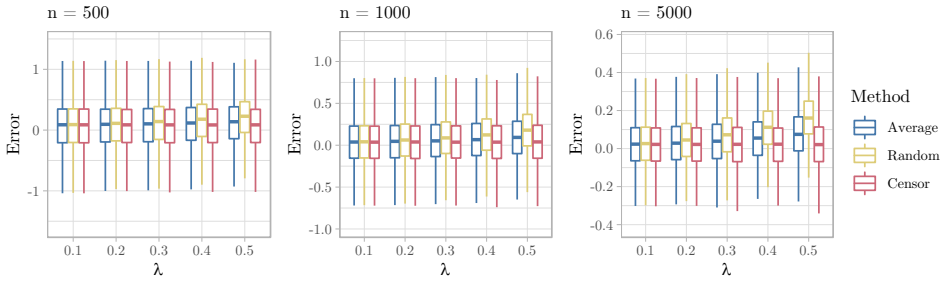


Figure 4.6: Boxplot of the estimation error $\theta - \hat{\theta}$ in the Joe copula with a percentage λ of ties in the lower tails generated symmetrically. In each margin, the percentage λ of the smallest samples are rounded to the first decimal, for Kendall's $\tau = 0.75$ with increasing severity.

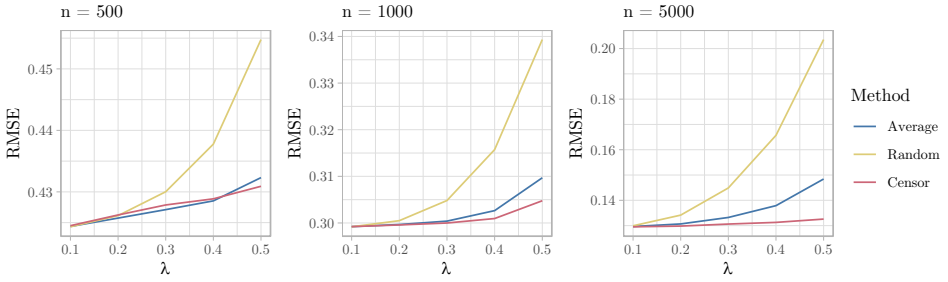


Figure 4.7: RMSE in the Joe copula with a percentage λ of ties in the lower tails generated symmetrically. In each margin, the percentage λ of the smallest samples are rounded to the first decimal, for Kendall's $\tau = 0.75$ with increasing severity.

Differences are more apparent for the Clayton copula. The error distribution and RMSE for the symmetric lower tail rounding experiment with $\tau = 0.25$ are shown in Figures 4.8 and 4.9. Here estimation by average ranks overestimate the correlation, while random ranks underestimate, whereas the interval censored estimation appears unbiased. For $\tau = 0.75$, plots of the RMSE and error distribution for the asymmetric lower tail rounding experiment are shown in Figures 4.10 and 4.11. Similar to the binning experiment, when correlations are strong, average and random ranks *consistently* underestimate the correlation, whereas interval censoring is accurate. Furthermore, the RMSE is stable regardless of the severity in ties. It is expected that the differences are greater in for the Clayton copula compared to the Joe copula, since the key feature of the copula, the lower tail, is censored.

In summary, interval censoring of bivariate copulae has better performance than estimation by random and average ranks, and the effect increases with both correlation and sample size. The method also seems to be unbiased, whereas random and average ranks are not. There are cases where the performance of the methods is comparable, however,

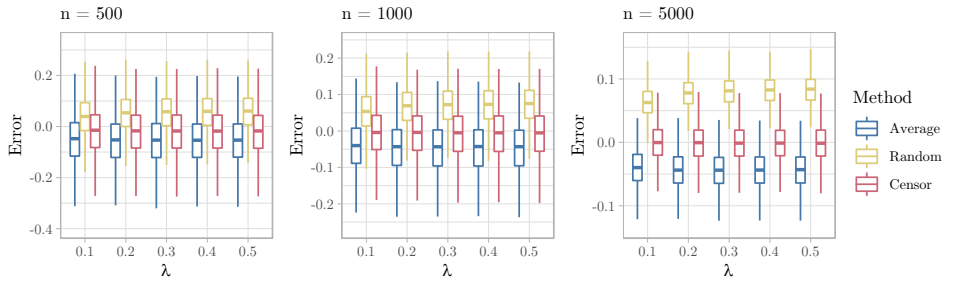


Figure 4.8: Boxplot of the estimation error $\theta - \hat{\theta}$ in the Clayton copula with a percentage λ of ties in the lower tails generated symmetrically. In each margin, the percentage λ of the smallest samples are rounded to the first decimal, for Kendall's $\tau = 0.25$ with increasing severity.

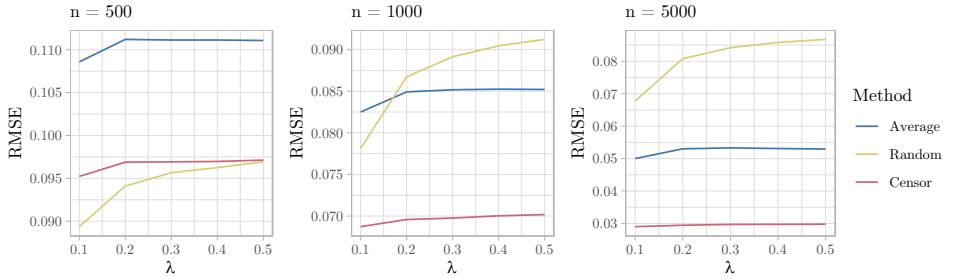


Figure 4.9: RMSE in the Clayton copula with a percentage λ of ties in the lower tails generated symmetrically. In each margin, the percentage λ of the smallest samples are rounded to the first decimal, for Kendall's $\tau = 0.25$ with increasing severity.

interval censoring is more robust against all occurrences of ties. In particular in censoring of key features of the copula, such as the tail dependence. Now we want to see if interval censored vine copulae can account for bias in larger multivariate models.

4.4 Vine Models

Now that interval censoring has shown evidence of unbiasedness for the smaller bivariate case, we wish to test if the methods proposed in Section 3.6.3 for constructing interval censored vines are equally effective. Two methods were proposed and both will be tested here. The simple extension (3.40) will be denoted "Censor", or *simple censoring*, and the other method (3.44) "CensorFull" or *full censoring*. Here we conduct similar experiments as in the previous section, that is, with both binned and lower tail rounded ties generated symmetrically and asymmetrically. The effectiveness of estimation is measured by introducing ties to samples collected from a given vine copula, and then estimating all parameters for this given vine, using different methods. This experiment has emphasis on whether the estimation methods are unbiased, so the structure of the vine is always given. As mentioned in the start of this chapter, such experiments are hard to manage when selection of

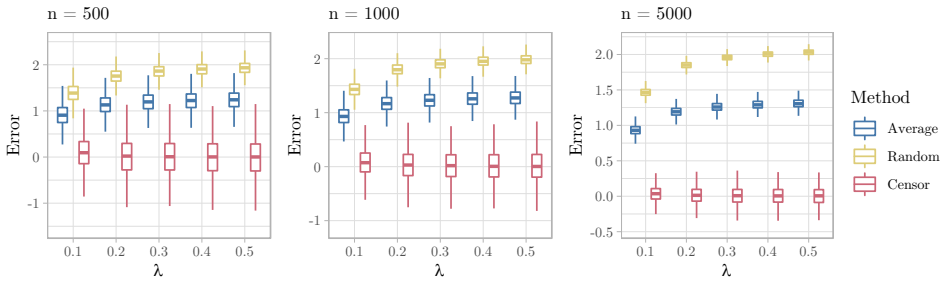


Figure 4.10: Boxplot of the estimation error $\theta - \hat{\theta}$ in the Clayton copula with a percentage λ of ties in the lower tails generated asymmetrically. The percentage λ of the smallest samples are rounded to the first decimal in the first margin, and to the second decimal in the second margin, for Kendall's $\tau = 0.75$ with increasing severity.

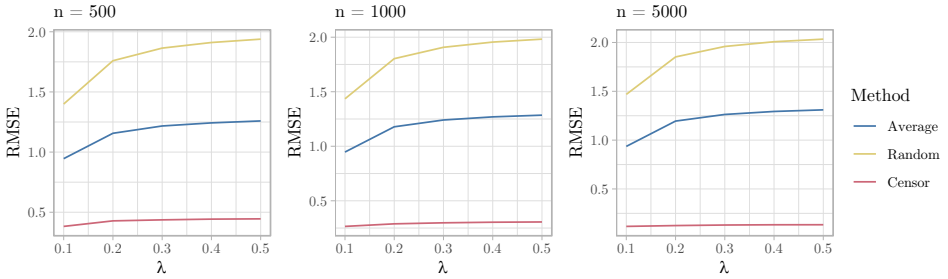


Figure 4.11: RMSE in the Clayton copula with a percentage of ties λ in the lower tails generated asymmetrically. The percentage λ of the smallest samples are rounded to the first decimal in the first margin, and to the second decimal in the second margin, for Kendall's $\tau = 0.75$ with increasing severity.

vine structure and copulae are also included. See Appendix A.1 for small experiment on bivariate copula selection.

For a vine copula, it is less meaningful to have all the copula parameters computed from the same correlation τ. By construction, vines typically have stronger correlations in the first tree. Therefore, we construct the vine with strongest correlations in the first tree, and weaker in the sequential trees. Interval censoring was highly effective against ties in the tails, and since modelling of asymmetric tail dependence is a key use case for vines, we include some copulae with tail dependence in the study. The structure is shown in Figure 4.12. Here $\tau \in \{0.3, 0.4, \ldots, 0.9\}$ denotes a base level that maintains the dependence of each bivariate copula, which is used to compute each copula parameter θ. The method used to apply ties to each margin is displayed in Table 4.1.

The vine construction appears to reduce the asymptotic efficiency of the estimation, in particular for strong correlations, which was also found in Haff (2012). Hence, reference solutions are also shown. That is, the model parameters are also estimated before introducing ties in the data, to separate the model impreciseness and the tie induced bias. In

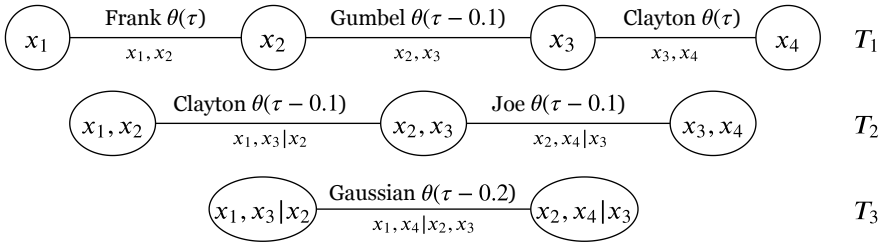


Figure 4.12: Overview of the vine copula used to generate samples. The correlations, τ, are used to compute the copula parameter θ for each bivariate copula. For all vine experiments, the structure is given, only parameter estimates are computed.

Symmetry	Experiment	x_1	x_2	x_3	x_4
Symmetric	Binning	$b = 15$	$b = 15$	$b = 15$	$b = 15$
	Tail rounding	1 DP	1 DP	1 DP	1 DP
Asymmetric	Binning	-	$b = 15$	$b = 15$	$b = 30$
	Tail rounding	-	1 DP	1 DP	2 DP

Table 4.1: Overview of the tie generating actions in each margin for the vine experiments. b shows the number of bins, DP (Decimal Place) denotes rounding to a decimal place, and - denotes no action.

this section, we discuss the results for the sample size $n = 5000$, the remaining results can be found in Appendix A. The differences are most apparent for large sample sizes, and the model error is smallest. In Section 4.4.1 we show the binning experiments, and in Section 4.4.2 the lower tail rounding experiments. For these sections, the estimation is only performed sequentially, as described in Section 3.4. In Section 4.4.3, we also conduct an experiment with joint estimation of all parameters, as introduced in Section 3.4.1, which in Haff (2012) was found to improve the asymptotic efficiency for strong correlations.

As mentioned in Section 4.1, the interval censored log-likelihood is harder to optimize than the average and random rank alternatives. This can occasionally cause the optimization to fail, in particular in the trees T_2 and T_3 when the data is severely tied and correlations are strong. For the error plots in this section, we have removed outliers that are caused by optimization issues. In some cases, there may also be general optimization issues, which is highlighted by the reference solution. This is likely due to the numerical approximations of the Hessian and gradient.

4.4.1 Binned Experiments

In this section we discuss the binning experiments defined in Table 4.1. Figure 4.13 shows the distributions of the estimation error for the symmetric experiment, while Figure 4.14 shows these distributions for the asymmetric experiment. For the first tree, the average and random ranks underestimate the correlation when it is strong, while interval censoring is unbiased, which are similar results to the bivariate case. For the second and third trees, however, *all* methods underestimate the correlation. The full censoring appears to

perform better in most cases, but it is not clearly unbiased, and generally underestimates the strength of correlation in comparison with the reference solution. The greatest improvement from interval censoring can be seen for T_2 in Figure 4.14, where full censoring performs significantly better than other methods. In T_3, however, the results are varying. The average rank error is comparable to full censoring for weak correlations, and smaller when the correlations is strongest. The latter may be due to optimization issues, however. In T_3 simple censoring has the smallest errors when $\tau = 0.6$, but will in the T_2 Clayton copula always fail with $\tau = 0.8$, thus always failing when $\tau = 0.7$ in T_3 also.

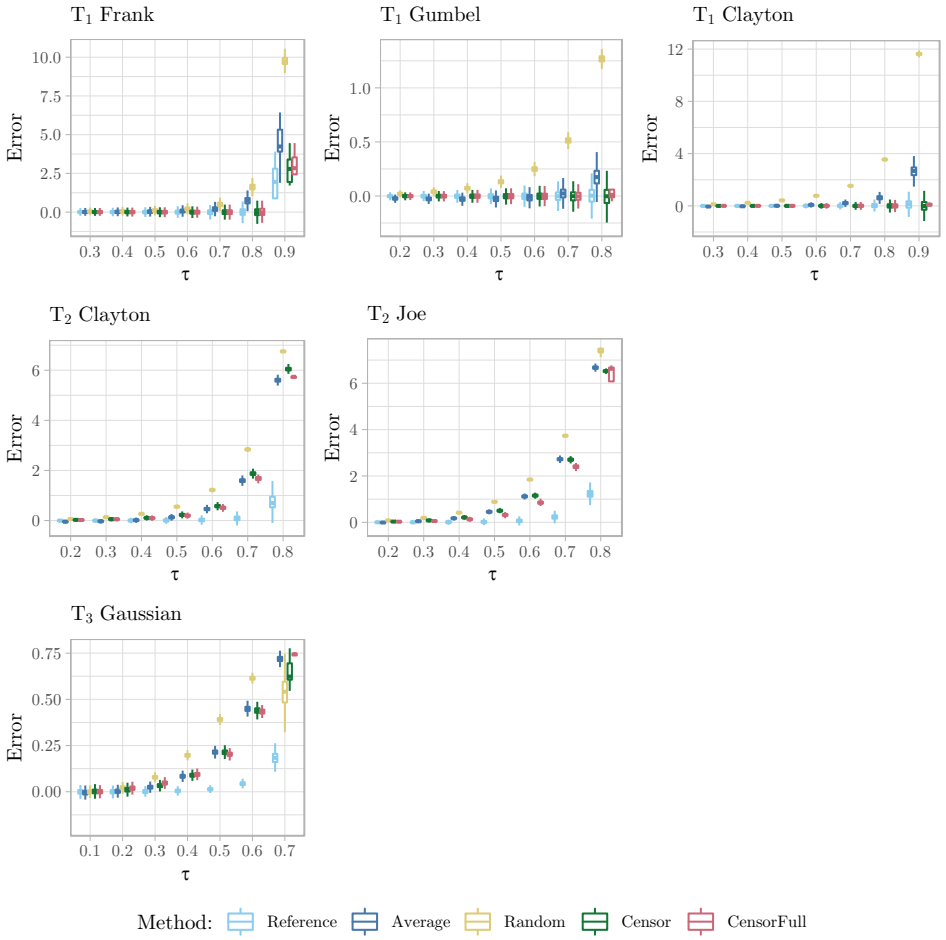


Figure 4.13: Boxplot of the estimation error $\theta - \hat{\theta}$ in the symmetrically binned vine copula of Figure 4.12. All four margins are binned with $b = 15$. The estimations are based on $n = 5000$ samples, and the vine parameters θ are computed from an increasing dependence τ.

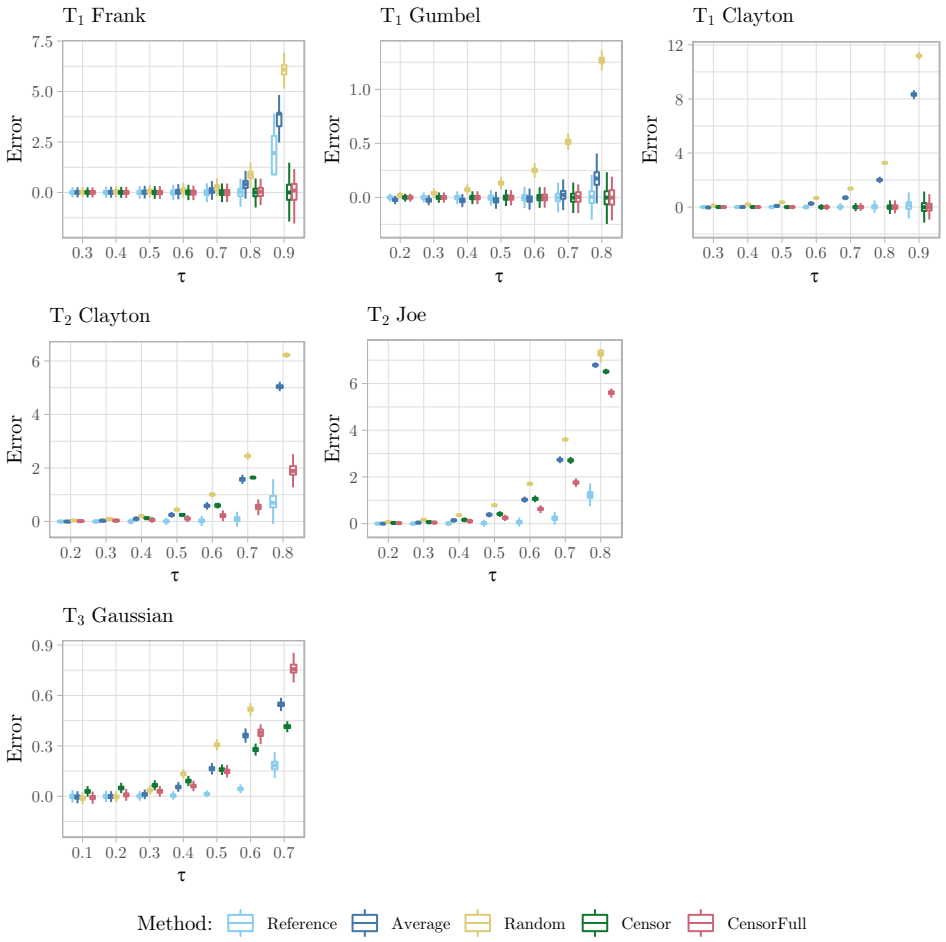


Figure 4.14: Boxplot of the estimation error $\theta - \hat{\theta}$ in the asymmetrically binned vine copula of Figure 4.12. The first margin is not tied, the second and third margins are binned with $b_{2,3} = 15$, and the fourth with $b_4 = 30$. The estimations are based on $n = 5000$ samples, and the vine parameters θ are computed from an increasing dependence τ. Notice that simple censored estimation always fails in the T_2 Clayton copula when $\tau = 0.8$.

4.4.2 Lower Tail Rounding

In this section we discuss the lower tail rounding experiments shown in Table 4.1. Firstly, when the base correlation is $\tau = 0.25$. Note that the first axis now reflects the severity of ties. The symmetric experiment is shown in Figure 4.15 and the asymmetric experiment in Figure 4.16. Similar to the binning experiments, the interval censored parameter estimates are accurate for the first tree, but generally underestimated in the second an third tree. However, full censoring gives accurate estimates for the T_2 copulae in both the symmetric and asymmetric experiment. Moreover, the simple censoring slightly underestimates

the correlation, whereas average and random ranks generally under and overestimate the strength of correlation. For the T_3 Gaussian copula, the censoring methods differ. The full censoring overestimates the correlation, while the simple censoring underestimates the correlation. Here, estimation by average ranks is closest, but not by much.

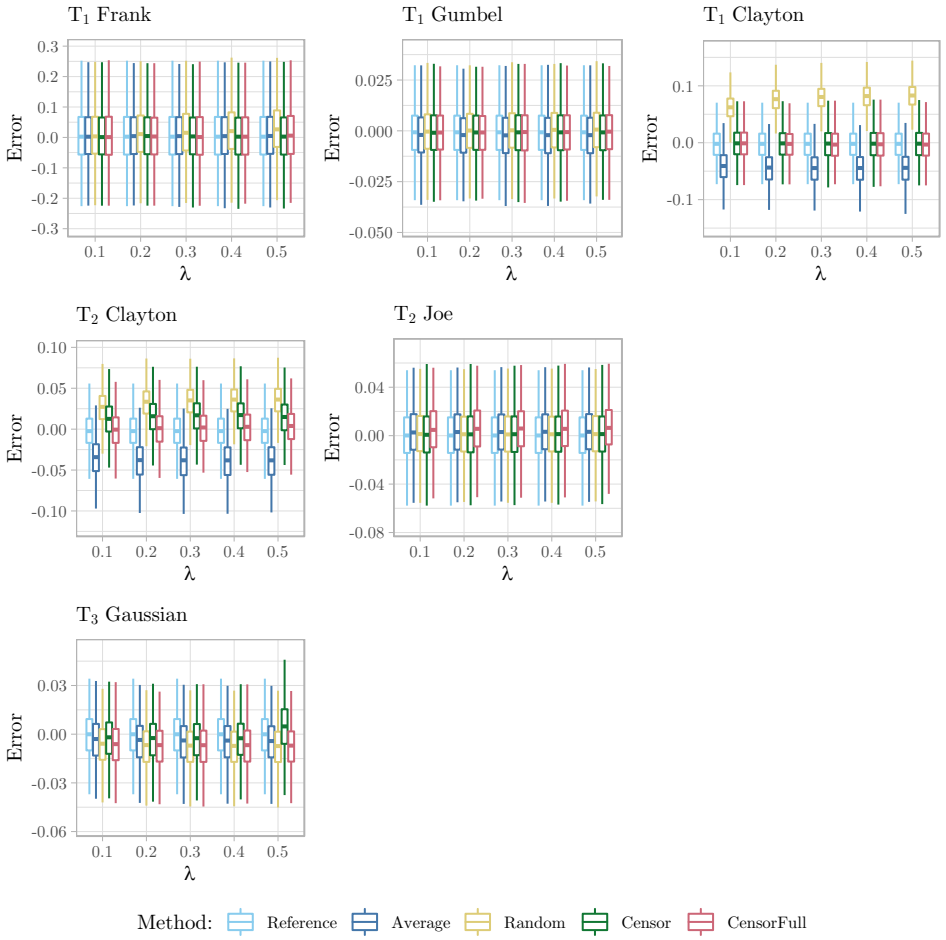


Figure 4.15: Boxplot of the estimation error $\theta - \hat{\theta}$ in the vine copula 4.12 with ties generated symmetrically in the lower tails. In each margin, the percentage λ of the smallest samples are rounded to the first decimal with increasing severity. The estimations are based on $n = 5000$ samples, and the vine parameters θ are computed from a base dependence $\tau = 0.25$.

Figures 4.17 and 4.18 show the errors when the strength of correlation increases to $\tau = 0.75$, and the first tree estimation is still unbiased when interval censoring. In T_2 for both the symmetric and asymmetric experiments, the strength of correlation is generally underestimated by all methods, but a little less for full censoring. The performance of the censoring and average methods are comparable, whereas random ranks underestimate

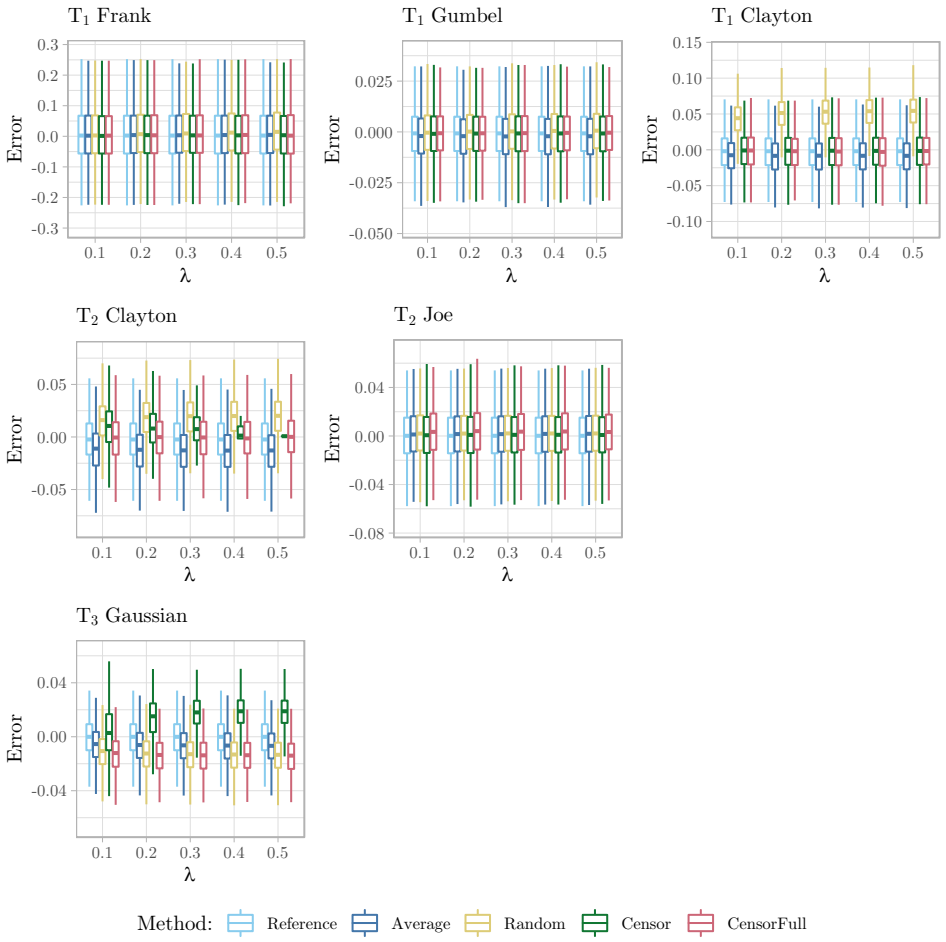


Figure 4.16: Boxplot of the estimation error $\theta - \hat{\theta}$ in the vine copula 4.12 with ties generated asymmetrically in the lower tails. The percentage λ of the smallest samples are rounded to the first decimal place in the second and third margin, and to the second decimal place in the fourth margin. The first margin is not rounded. Estimations are based on $n = 5000$ samples, and the vine parameters θ are computed from a base dependence $\tau = 0.25$.

even more. In the asymmetric experiment, full censoring almost achieves an accurate estimate for both copulae in T_2. The correlation in the T_3 Gaussian is underestimated in the symmetrical case, but in the asymmetrical experiment both censoring methods are closer than the reference solution. In the previous experiment, when $\tau = 0.25$, full censoring underestimated the correlation in the T_3 Gaussian copula, hence, it is possible that some estimation error accounts for the model induced error.

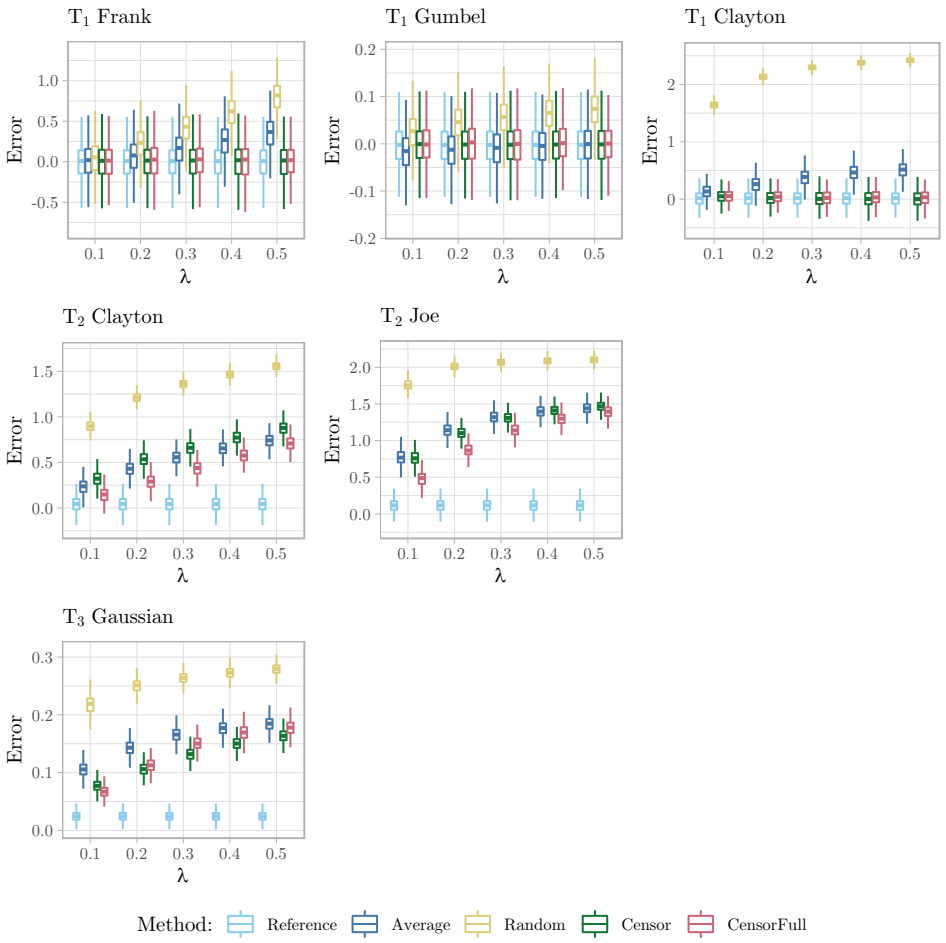


Figure 4.17: Boxplot of the estimation error $\theta - \hat{\theta}$ in the vine copula 4.12 with ties generated symmetrically in the lower tails. In each margin, the percentage λ of the smallest samples are rounded to the first decimal with increasing severity. The estimations are based on $n = 5000$ samples, and the vine parameters θ are computed from a base dependence $\tau = 0.75$.

4.4.3 Joint Estimation

In Haff (2012), a sequential estimation followed by a joint was found to improve the efficiency. Joint interval censored estimation is very time consuming, so we only perform one of the previous experiments, which is the symmetric binning experiment described in Table 4.1. To speed up computations even more, the T_3 Gaussian copula in Figure 4.12 is exchanged for a Frank Copula, since estimation of the one parameter Archimedean copulae is generally faster than the Gaussian. The new vine is shown in Figure 4.19, and the estimation errors are shown in Figure 4.20. It is apparent that the full censored estimation is unstable, by the increased estimation variance. As before, interval censored estimation

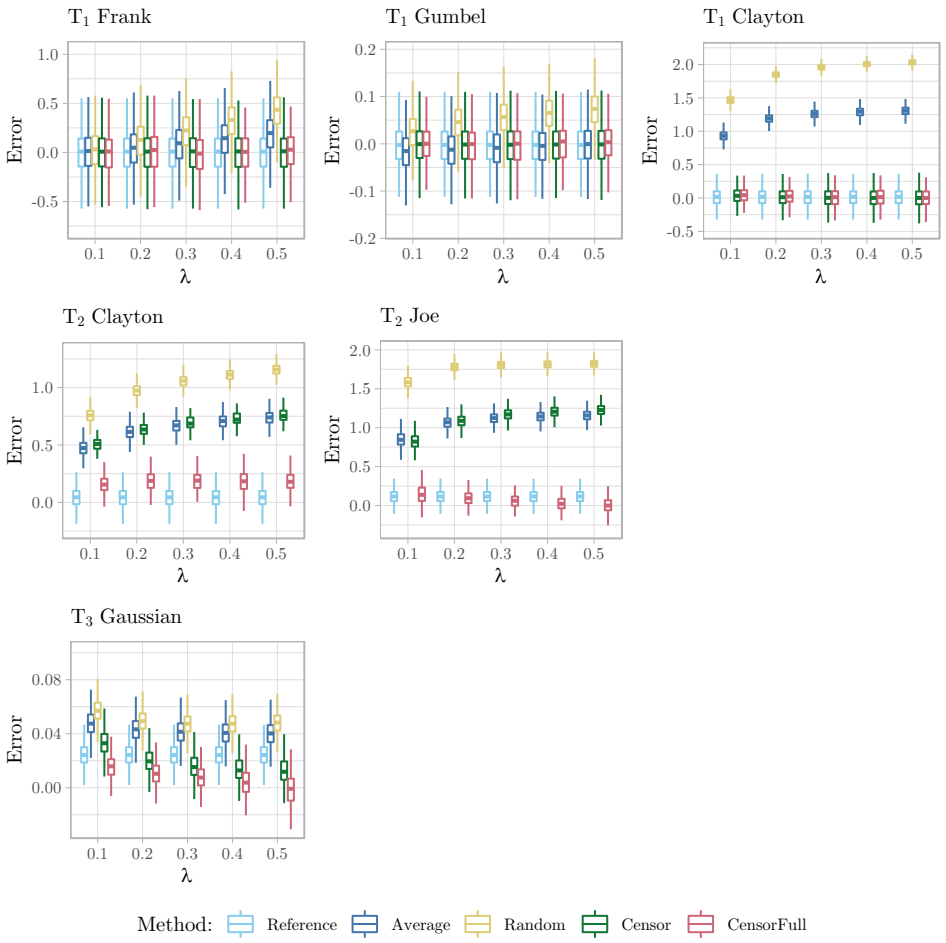


Figure 4.18: Boxplot of the estimation error $\theta - \hat{\theta}$ in the vine copula 4.12 with ties generated asymmetrically in the lower tails. The percentage λ of the smallest samples are rounded to the first decimal place in the second and third margin, and to the second decimal place in the fourth margin. The first margin is not rounded. Estimations are based on $n = 5000$ samples, and the vine parameters θ are computed from a base dependence $\tau = 0.75$.

is unbiased in tree one. Estimation by average ranks, however, will generally achieve a closer estimate than the censoring methods in the second and third tree, though, not by much. This is likely due to optimization issues, as indicated by the inconsistent estimates of the full censoring scheme. In fact, the full censoring scheme has an approximate fail rate of 60% and the simple censoring a rate of 47%. It appears that numerical approximations of the gradient and hessian are not sufficient for joint optimization. The joint optimization is also very time consuming, and can take up to one hour when $n = 5000$.

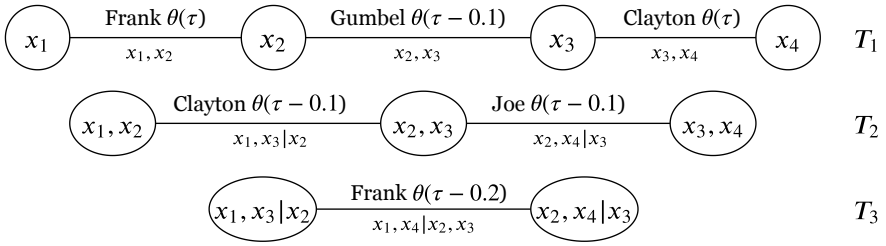


Figure 4.19: Overview of the vine copula used to generate samples for the joint estimation experiment. The correlations, τ, are used to compute the copula parameter θ for each bivariate copula. Only parameter estimates are computed for the given structure.

4.4.4 Summary

In summary, interval censored estimation is unbiased in the bivariate case, but not generally for vine copulae. Despite the poor result in comparison with bivariate censoring, we still advocate the use of interval censored vines, and specifically by the full censoring method. The estimation of the first tree is generally unbiased, and compared with other methods, the performance is better for sequential trees. Furthermore, tied lower tails are commonly found in precipitation event data, which, as demonstrated, is a good use case for interval censoring. It should also be noted that the issues are most severe when correlations in later trees are high. This is not typically the case for real data. It was suspected that the results for T_2 and T_3 would improve significantly if correlations were weaker in these trees. That is, by using the same correlations in T_1, and even weaker in T_2 and T_3. This appears not to be the case, and a small example can be seen in Appendix A.2.

There are a few other remarks to be noted. Firstly, the estimation will sometimes return unreasonable results, which have been removed in the plots. We computed over 1200 repeated experiments, so the results should sill give an indication of the validity of the methods. Secondly, the interval censored maximum likelihood optimization is more difficult in comparison with other methods. The optimization is based on numerical approximations of the gradient and hessian, which is slow. A joint optimization of all vine parameters might be more accurate and feasible following optimization improvements.

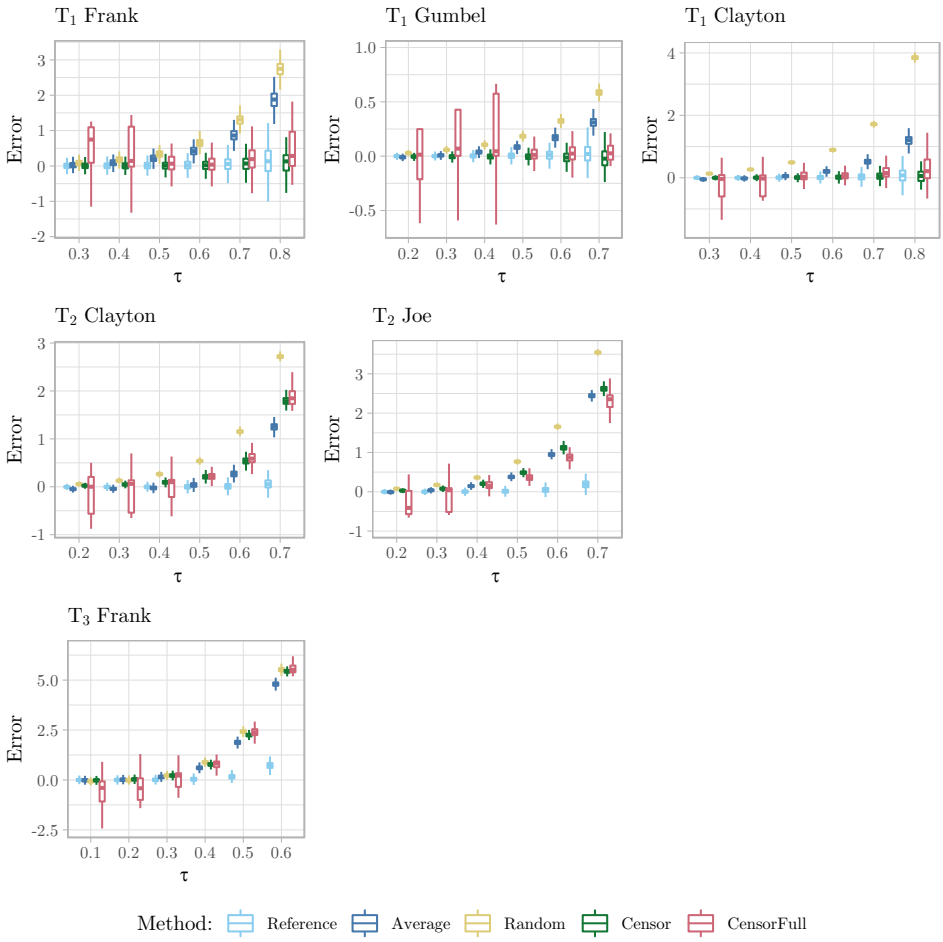


Figure 4.20: Boxplot of the estimation error $\theta - \hat{\theta}$ in the symmetrically binned vine copula of Figure 4.19. All four margins are binned with $b = 15$. The estimations are based on $n = 5000$ samples, and the vine parameters θ are computed from an increasing dependence τ. The estimation is first performed sequentially, and then jointly over all vine parameters.

Chapter 5

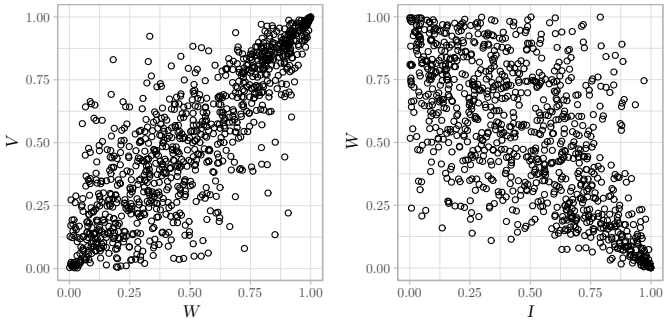
Copula Modelling for Precipitation and Temperature Data

In this chapter, we first give a short summary of the data quality in regards to copula modelling, in Section 5.1. In order to use vine copulae to construct a joint distribution for the event parameters, however, the data has to satisfy model assumptions. This is done in Section 5.2 by applying more formal hypothesis tests, followed by an application of the theory of time series to find a suitable model for temperature. The chosen model is then tested for underlying assumptions. Finally, in section 5.3 we conclude by building two interval censored regular vine models from the characteristic parameters of an event and the selected temperature model. The models are validated by goodness-of-fit testing of each bivariate copula in the vines. The modelling of temperature and precipitation events is intended to show the steps that were taken before ultimately deciding that the results are of minor interest, due to the poor data quality and general challenges with the event model. Furthermore, the main topic for this study is in copula modelling, and not the marginal distributions. This was done in Birketvedt (2019), where we also gave a more elaborate discussion of data characteristics for precipitation events.

5.1 Data Summary

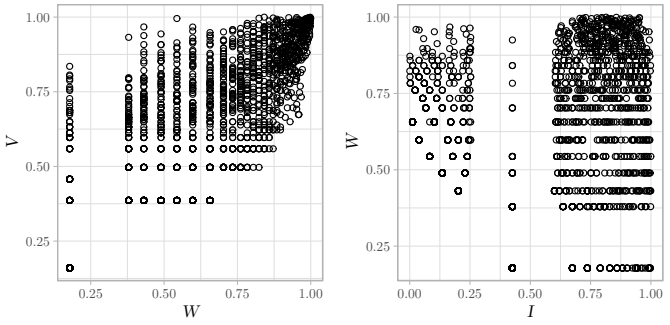
The essence of copula modelling is that we can divide the modelling of a large multivariate distribution into two separate cases: the joint behaviour and marginal behaviour. The joint behaviour is modelled by the copula, which is unique for *continuous* marginal distributions. The pseudo-observations are estimations of the marginal distributions, and jointly they give an indication of the shape of the underlying copula. Ideally these should look similar to the samples in Figure 5.1. However, before we apply a threshold for rain volume V for the data at Særheim the pseudo-observations are presented in Figure 5.2. For low values both W and W show large gaps in the estimated marginals. This is a consequence of ties, which clearly distort the underlying copula, and the process does not look continu-

ous. As mentioned, this can partly be resolved by filtering with some threshold (Salvadori & Michele, 2006). We used $V > 1$ mm. The resulting pseudo-observations are shown in Figure 5.3. The data now resemble the desired observations in 5.1, but is still clearly distorted by ties. In particular for the duration W, which appears in "layers".



(a) **(b)**

Figure 5.1: An illustration of ideal pseudo-observations.

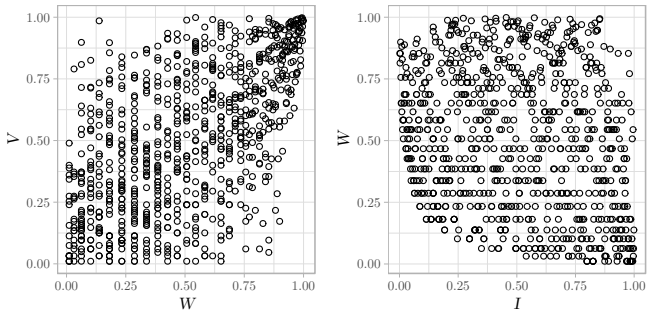


(a) (W, V) **(b)** (I, W)

Figure 5.2: Unfiltered pseudo-observations for (I, W) and (W, V) from Særheim in summer.

5.2 Selection of Temperature Model

The goal of this section is to include a measure for temperature in the precipitation event model. Ideally, this measure should be strongly correlated with the other event parameters, in particular intensity, and also form i.i.d. events. In section 5.2.1, the i.i.d. assumption of the current event model (or standard event model), is evaluated by hypothesis testing of stationarity and multiple serial independence. Temperature appears to have a seasonal trend, which is accounted for in Section 5.2.2 with a time series modelling of candidate temperature parameters. Each candidate parameter is then included in the event model,



(a) (W, V) **(b)** (I, W)

Figure 5.3: Filtered pseudo-observations for (I, W) and (W, V) from Særheim in summer. $V > 1$ mm.

and tested for stationarity and serial independence, to check whether the full model is i.i.d. In this section we will refer to the event model for the parameters I, W, V and D as the standard event model, and temperature-event model when temperature is included.

5.2.1 Stationarity of the Events Model

In this section, we apply the tests from Section 3.7 to test the current event model for stationarity and multiple serial independence. In order to verify that the full temperature model is sufficiently i.i.d, the underlying model based on volume V, duration W, intensity $I = V/W$ and the dry period D must also satisfy these requirements. First we test for stationarity in the multivariate distribution by applying the method of change point detection (3.48). This test is implemented in R-package `npcp` (Kojadinovic, 2017) and was executed with the automation procedure described in Bücher & Kojadinovic (2016), and the fast resampling scheme in Bücher et al. (2014). The resulting p-values are shown in Table 5.1. At a significance level of $\alpha = 0.05$, only the seasons winter and spring can be considered stationary. Since the seasonal separation is done by dividing based on the calendar year, and not weather specifically, a more hydrologically motivated separation could potentially give a better separation of the different distributions.

Season	p-Value
Winter	0.1843
Spring	0.0714
Summer	0.0065
Fall	0.0045

Table 5.1: Resulting p-values from a test of change point detection in multivariate stationarity of the standard event model for all seasons.

The aim of this study is to model the underlying copula, so while the events are not found to have stationarity in the multivariate distribution, we still apply the test (3.53),

which emphasizes changes in the underlying copula given stationarity in the marginal distribution. An implementation for this test is also available in `npcp`, and was conducted similar to the previous test. Table 5.2 shows the resulting p-values. Here all the underlying copulae are found to be stationary for all seasons at a significance level $\alpha = 0.05$.

Season	p-Value
Winter	0.3012
Spring	0.7697
Summer	0.1813
Fall	0.1234

Table 5.2: Resulting p-values from a test of change point detection multivariate stationarity in the underlying copula of the standard event model for all seasons.

Finally, we apply the test of multiple serial independence, both globally (3.55) and by the Möbius-decomposition (3.56). In Kojadinovic & Yan (2011), the authors found that for a given dimension, the power of the test decreases as the embedding dimension p increases, however, finding a good choice for p is still difficult. Computing the test for large values of p is also costly, due to the many subsets generated from the Möbius decomposition. Therefore, we compute the test for $p = 14$ and a maximum subset size $m = 3$, and for $p = 4$ with no subset restrictions, and check whether the tests are in accordance. The results are presented in Table 5.3, and the events are generally found to be serially independent, with the exception of p-values from Fisher's rule in summer. At a significance level of $\alpha = 0.05$, both test specifications give similar results, with the exception of Tippett's rule in summer, which indicates serial independence for $p = 4$, and dependence for $p = 14$. The global test indicates independence, however.

Season	p	Global	Fisher	Tippett	p	Global	Fisher	Tippett
Winter	14	0.3781	0.5809	0.8487	4	0.5300	0.7298	0.8636
Spring		0.2223	0.4221	0.5829		0.4231	0.6289	0.6958
Summer		0.3102	0.0165	0.0005		0.0644	0.0315	0.2373
Fall		0.2023	0.2203	0.6269		0.5739	0.4740	0.6868

Table 5.3: Resulting p-values from a test of multiple serial independence in of the standard event model for all seasons.

In summary, all tests are not passed for all seasons, only for winter and spring. These tests are, however, more rigorous than the tests or discussions used in similar event modelling attempts, such as Vandenberghe et al. (2010); De Michele & Salvadori (2003); Salvadori & Michele (2007), which can be regarded as successful. For this study, regardless, an important purpose of these tests is to judge whether the inclusion of the candidate temperature parameters affect the stationarity and serial independence, and not the other parameters.

5.2.2 Time Series Modelling of Temperature

In this section, we find a time series model for the candidate temperature parameters mean temperature T, mean dry temperature T_D, maximum temperature T_M, minimum temperature T_m, maximum dry temperature T_{DM}, minimum dry temperature T_{DM}, temperature difference T_Δ and dry temperature difference $T_{D\Delta}$, as introduced in chapter 2. Figure 5.4 shows the mean temperature of the events for each day of the year, and there is a clear seasonal trend which has to be removed. This is done by applying the time series analysis described in section 3.7.

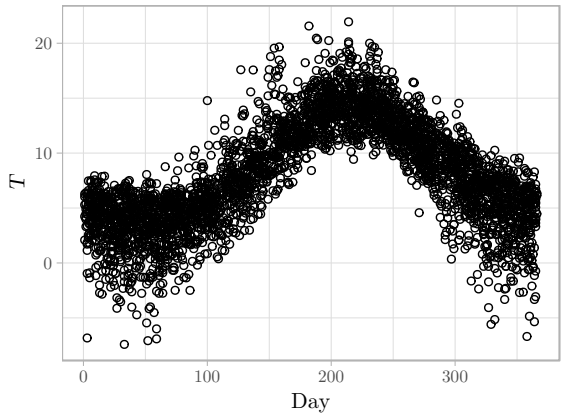


Figure 5.4: Mean temperature T of the events at Særheim for each day of the year.

The emphasis of this study is not primarily time series modelling, but precipitation modelling by copula. Therefore, we attempt a set of methods, and choose the one that removes trend and serial correlations best. A more advanced model is not straight forward, since the events form an irregular time series. That is, the observations are not equally spaced, but separated by varying time gaps. Keeping this in mind, the following model is fitted to all temperature parameters

$$\begin{aligned}\boldsymbol{Y}_i =& a + \sum_{k=1}^{K} \left[\alpha_{K,k} \sin(2\pi ki/365) + \beta_{K,k} \cos(2\pi ki/365)\right] + \\ & \sum_{m=1}^{M} \left[\alpha_{M,m} \sin(2\pi mi/24) + \beta_{M,m} \cos(2\pi mi/24)\right] + N_i,\end{aligned}$$

where N_i is the ARIMA-model (3.66). The best fit is selected from minimal AIC in a search with ARIMA parameter restriction $(p \leq 10, d \leq 2, q \leq 10)$ and $K, M \leq 5$. Notice that we have fitted a model with Fourier terms as external regressors, which possibly avoids some issues that arise from the irregular time series. The ARIMA-model is intended to primarily correct possible serial correlations. Since trend is described as a function of day and hour, we avoid interpolation of an ARIMA-model. Furthermore, Fourier terms are able to model complex seasonal pattern, which in this context is that temperature is a

combination of both season and time of day. Note that the standard ARIMA-model N_i is also included in the search, that is, without the Fourier-terms as external regressors. For T, T_D, T_M, T_m, T_{DM} and T_{Dm} the typical models are ARIMA($2 \leq p \leq 4, d = 0, 1 \leq q \leq 2$) and $0 \leq K \leq 5, 1 \leq M \leq 2$, whereas the parameter T_Δ results in the model ARIMA($p = 5, d = 1, q = 0$), and $T_{D\Delta}$ gives ARIMA($p = 0, d = 1, q = 1$). Note that $K = 0$ denotes removal of the terms $\alpha_{K,k}$ and $\beta_{K,k}$. Figure 5.5 shows the residuals from the ARIMA-model for mean temperature T. The residuals show no clear signs of trend, but there are some outliers.

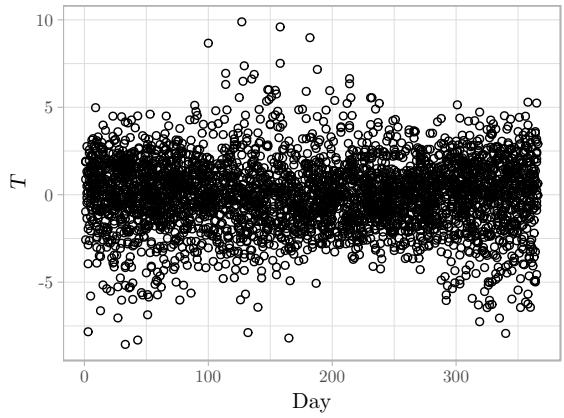


Figure 5.5: Mean temperature T residuals from the ARIMA-model.

A quick test for goodness-of-fit of an ARIMA-model, is the Ljung-Box test described in Section 3.7.4. We test the residuals for different maximum lags. T_Δ is *not* found significant for any maximum lag, T_m is significant up to a maximum lag of 10, whereas the remaining parameters are significant up to a maximum lag of 14. Such Box tests are generally weaker than the test based on the sequential empirical copula process (Bücher et al., 2014), and it is of greater interest to test for sufficient stationarity and serial independence in the multivariate event model. Hence, we conduct all the tests in Section 5.2.1 with the inclusion of each individual temperature parameter.

The p-values for the test of change point detection in multivariate stationarity are presented in Table 5.4. The seasons winter and spring are generally found to be stationary for all parameters at a significance level of $\alpha = 0.05$, whereas the events in summer and fall are not.

The p-values from test for multivariate stationarity in the underlying copula are shown in Table 5.5, and here all models are found significant at a level $\alpha = 0.05$.

Table 5.6 shows p-values from the test of multiple serial independence following Fisher's combination rule. The parameters generally show serial independence for all seasons, with the exception of summer where fewer models are found to be serially independent. Mean temperature T, max temperature T_M and min temperature T_m are not serially independent in summer.

The tests show similar results as before temperature was introduced. The minimum dry temperature T_{Dm} is an exception, though. After inclusion in summer, the model is

Season	T	T_D	T_M	T_m	T_{DM}	T_{Dm}	T_Δ	$T_{D\Delta}$
Winter	0.380	0.090	0.330	0.449	0.239	0.087	0.268	0.334
Spring	0.091	0.063	0.083	0.164	0.220	0.051	0.074	0.317
Summer	0.004	0.003	0.010	0.016	0.003	0.002	0.009	0.026
Fall	0.006	0.001	0.009	0.014	0.010	0.002	0.001	0.011

Table 5.4: Resulting p-values from a test of multivariate stationarity for all seasons with the inclusion of individual temperature parameters.

Season	T	T_D	T_M	T_m	T_{DM}	T_{Dm}	T_Δ	$T_{D\Delta}$
Winter	0.204	0.377	0.367	0.221	0.383	0.401	0.426	0.073
Spring	0.942	0.656	0.746	0.960	0.489	0.628	0.755	0.534
Summer	0.615	0.174	0.582	0.075	0.123	0.039	0.521	0.337
Fall	0.305	0.251	0.367	0.425	0.361	0.158	0.175	0.215

Table 5.5: Resulting p-values from a test of multivariate stationarity in the underlying copula for all seasons with the inclusion of individual temperature parameters.

no longer stationary in the underlying copula and serially independent. For the remaining temperature parameters, the results are generally the same, so the suggested time series model seems successful. The serial dependence and multiple serial dependence seem to be more dependent on the seasonal separation of events, and not the ARIMA-model and temperature parameter choice. As mentioned, a better seasonal separation may give a better model. In Norway, for instance, the seasons may be separated based on thresholds for mean daily temperature Dannevig (2019).

5.3 Regular Vine Construction

In this section we construct two vine copula models for the precipitation events with the inclusion of temperature parameters. The choice of temperature parameters is based on the strongest correlation with the standard event parameters. We build one larger 6 parameter model to demonstrate the versatility, and one intensity-duration-temperature model which was the intended relationship for investigation in this study. Since the full censoring method (3.44) showed the best results in the simulation study in Chapter 4.4.4, we emphasize the results from this method here. The simple censoring method was also used (3.40), and the results are given in Appendix B, along with some of the full censoring results not presented here. In the full weather models, we emphasize different seasonal structures for the vines, while in the smaller models, we look at seasonal differences in dependence.

Table 5.7 shows the strongest correlations between the the event parameters, and the candidate temperature parameters. The most strongly correlated overall are the temperature difference parameters T_Δ and $T_{D\Delta}$. The intensity I is an exception, and has the strongest correlation with a different temperature parameter for all seasons. The correlation is weak, though, so this may be random. Intensity is most highly correlated with

Season	p	T	T_D	T_M	T_m	T_{DM}	T_{Dm}	T_Δ	$T_{D\Delta}$
Winter	14	0.061	0.050	0.124	0.197	0.473	0.019	0.468	0.619
Spring		0.694	0.455	0.871	0.612	0.972	0.282	0.947	0.773
Summer		0.020	0.058	0.032	0.023	0.146	0.050	0.113	0.220
Fall		0.415	0.093	0.509	0.080	0.196	0.024	0.733	0.832
Winter	4	0.143	0.047	0.148	0.544	0.443	0.018	0.602	0.468
Spring		0.234	0.326	0.494	0.304	0.698	0.235	0.971	0.963
Summer		0.029	0.144	0.042	0.034	0.201	0.070	0.217	0.115
Fall		0.724	0.556	0.641	0.522	0.534	0.509	0.865	0.650

Table 5.6: Resulting p-values from Fisher's combination rule in a test of multiple serial independence in the events for all seasons with the inclusion of individual temperature parameters.

temperature during winter, and the least during summer. Both temperature difference parameters are serially independent, and pass the test for serial independence in the underlying copula, thus, we choose these for the full weather model. For the smaller intensity-duration-temperature models, we choose the most strongly correlated temperature parameter for each season.

Season	Parameter	Temperature	Kendall's τ
Winter	V	T_Δ	0.337
	W	T_Δ	0.433
	I	T	0.180
	D	$T_{D\Delta}$	0.373
Spring	V	T_Δ	0.230
	W	T_Δ	0.402
	I	T_m	0.101
	D	$T_{D\Delta}$	0.446
Summer	V	T_Δ	0.278
	W	T_Δ	0.427
	I	T_D	0.058
	D	$T_{D\Delta}$	0.457
Fall	V	T_Δ	0.341
	W	T_Δ	0.474
	I	T_M	0.139
	D	$T_{D\Delta}$	0.413

Table 5.7: Candidate temperature parameters that are most strongly correlated with the event parameters for each season.

Before constructing the vine models, we perform the tests in Section 5.2.1 on the smaller event models. The larger models were too computationally demanding for these tests. Note that, the smaller model has a different temperature parameter for each season, which gives greater variability in the tests of stationarity and serial independence. At a significance level of $\alpha = 0.05$, the models fail stationarity in the multivariate distribution

for summer, (I, W, T_D), and multiple serial independence in winter, (I, W, T). The vines are selected following the sequential estimation procedure, described in Section 3.4, and each bivariate copula is tested for goodness-of-fit, by the test described in Section 3.6.2. The number of bootstraps are set to ten times the number of observations, as suggested in Genest et al. (2006).

Some of the optimization issues described in Chapter 4.4.4 are still present here. In particular, large upper optimization limits cause issues, even if the true value is small. As an example, the estimation of a Clayton copula with $\theta = 2$ could fail if $\theta = 25$ was set as upper limit, and converge with $\theta = 20$ as upper limit. Therefore, the optimization limits were determined such that computations ran smoothly. Note that even after the adjustments, all parameter estimates are significantly smaller than the upper limits. Furthermore, parametric bootstrapping requires some stability to obtain accurate p-values, and may still fail after the adjustments. Hence, we monitored the stability of the computations.

For the full weather model, the seasons winter, spring and summer have the same vine construction, while fall has a different one. Here we present the models for spring and fall. Figure 5.6 shows the first vine tree for spring and Figure 5.7 for fall. In the vine for spring, each node has two edges at max, and this is referred to as a D-vine. In fall the correlation between T_Δ and $T_{D\Delta}$ is weaker, and not joined by an edge, which gives the vine a different structure. However, the parameters are in general more strongly correlated in fall.

Tables 5.8 and 5.9 show the corresponding copulae with their p-values from the goodness-of-fit test. In fall all copulae in the trees $T_3, \ldots, T_5$ are the independence copula. For spring, there are a few significant copulae in these trees, but their correlations are weak. Both seasons have similar copulae for the same parameter pairs, with the exception of (W, T_Δ) and $(D, T_{D\Delta})$. In spring, (W, T_Δ) is modelled by a Gaussian copula, while it is modelled by a Frank copula in fall. Both are found significant. Furthermore, $(D, T_{D\Delta})$ is modelled by a Gaussian copula in spring, and a Gumbel copula in fall. The Gumbel copula is found to be significant, whereas the Gaussian is not. Notice also that only one-parameter copulae were selected here.

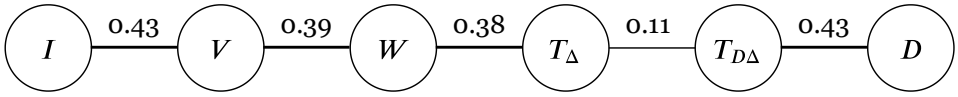


Figure 5.6: First tree in the vine for the full weather model in spring. Edges for significant copulae show Kendall's τ.

For the smaller models, summer and winter are presented. The vines are shown in Figures 5.8 and 5.9, and the corresponding copulae in Tables 5.10 and 5.11. Both vines have same structure, and same copula families in the first tree. Intensity-duration is modelled by a rotated Clayton copula, and intensity-temperature is modelled by Frank's family. The T_2 copula is different, however. In winter, the copula $C_{W,T|I}$ is modelled by a rotated Joe copula, whereas in summer, this pair is independent. It should be noted that the temperature parameters are different, which makes a direct comparison less interesting. All copulae in these models are found significant in goodness-of-fit testing.

Even if these seasons are modelled by the same families of copula, the structure is still

Tree	Copula	Family	θ	δ	p-value
T_1	$C_{V,I}$	Frank	4.53	–	0.00
	$C_{W,V}$	Gaussian	0.57	–	0.11
	$C_{T_\Delta,W}$	Gaussian	0.57	–	0.06
	$C_{T_{D\Delta},D}$	Gaussian	0.62	–	0.02
	$C_{T_\Delta,T_{D\Delta}}$	Gumbel	1.12	–	0.92
T_2	$C_{T_\Delta,D\|T_{D\Delta}}$	Gumbel 90°	1.06	–	0.90
	$C_{W,I\|V}$	Frank	−32.51	–	0.00
	$C_{T_\Delta,V\|W}$	Independence	–	–	–
	$C_{T_{D\Delta},W\|T_\Delta}$	Independence	–	–	–
T_4	$C_{V,D\|W,T_{D\Delta},T_\Delta}$	Frank	−0.51	–	0.68
	$C_{T_{D\Delta},I\|T_\Delta,V,W}$	Independence	–	–	–

Table 5.8: Full censoring of the full weather model for spring. Rotated copulae are shown by the following degree. θ denotes the first copula parameter, and δ the second. The remaining pairs are modelled by the independence copula.

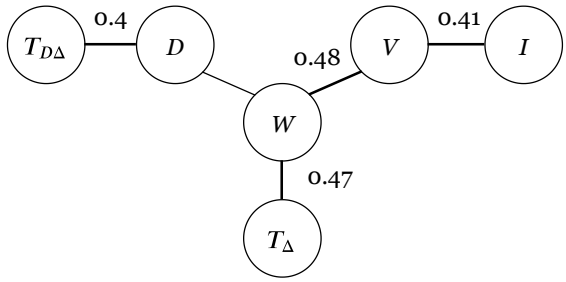


Figure 5.7: First tree in the vine for the full weather model in fall. Edges for significant copulae show Kendall's τ.

different. Figures 5.10 and 5.11 show density plots of the fitted copulae for each season. The relationship between intensity and duration is more strongly correlated in summer than winter. In contrast, the relationship between temperature and intensity is stronger in winter. Something to note is that in the current rotation, the 90° Clayton copula models lower right tail dependence. The implication of this is that high intensity precipitation will generally have short duration, and this effect increases with higher intensities. This finding is similar to Birketvedt (2019). The Frank copula for intensity-temperature, however, is not able to model tail dependence.

We also give a few general remarks. The pair (I, V) is modelled by a Frank copula for all seasons, see Appendix B.1 for details. In all seasons, it fails the goodness-of-fit test. However, the pair (I, W) is successfully modelled by a 90° Clayton copula for all seasons. In Birketvedt (2019), the Clayton copula was selected for this relationship in winter, but not found significant in goodness-of-fit testing. It seems that estimation and inference procedures are improved by interval censored estimation and bootstrapping. The pair (W, V) is modelled by a Gaussian copula in spring and fall, and by a Gubmbel copula

Tree	Copula	Family	θ	δ	p-value
T_1	$C_{I,V}$	Frank	4.34	–	0.00
	$C_{W,V}$	Gaussian	0.69	–	0.01
	C_{W,T_Δ}	Frank	5.18	–	0.27
	$C_{D,T_{D\Delta}}$	Gumbel	1.66	–	0.96
	$C_{D,W}$	Independence	–	–	–
T_2	$C_{I,W\|V}$	Gumbel 90°	12.08	–	0.01

Table 5.9: Full censoring of the full weather model for fall. Rotated copulae are shown by the following degree. θ denotes the first copula parameter, and δ the second. The remaining pairs are modelled by the independence copula.

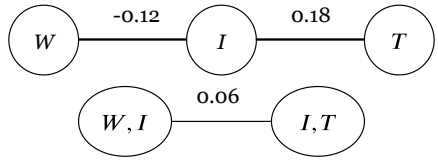


Figure 5.8: Vine for the intensity-duration-temperature model in winter. Edges for significant copulae show Kendall's τ.

in winter and summer. The p-values for the Gumbel copula are, however, significantly larger than for the Gaussian. There is also only one two-parameter copula selected, which is the 180° BB8 copula for $C_{W,I|V}$ in winter.

Tree	Copula	Family	θ	δ	p-value
T_1	$C_{I,W}$	Clayton 90°	0.26	–	0.99
	$C_{I,T}$	Frank	1.67	–	0.11
T_2	$C_{W,T\|I}$	Joe 180°	1.12	–	0.90

Table 5.10: Full censoring of the small weather model for winter. Rotated copulae are shown by the following degree. θ denotes the first copula parameter, and δ the second.

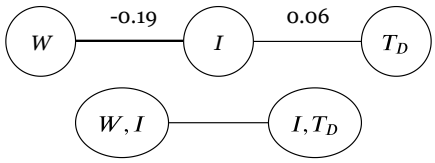
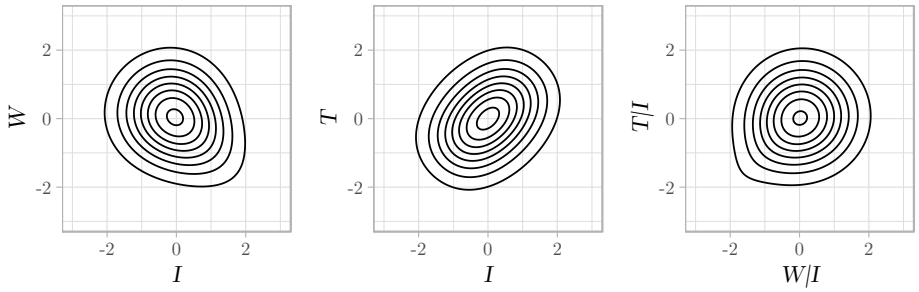


Figure 5.9: Vine for the intensity-duration-temperature model in summer. Edges for significant copulae show Kendall's τ.

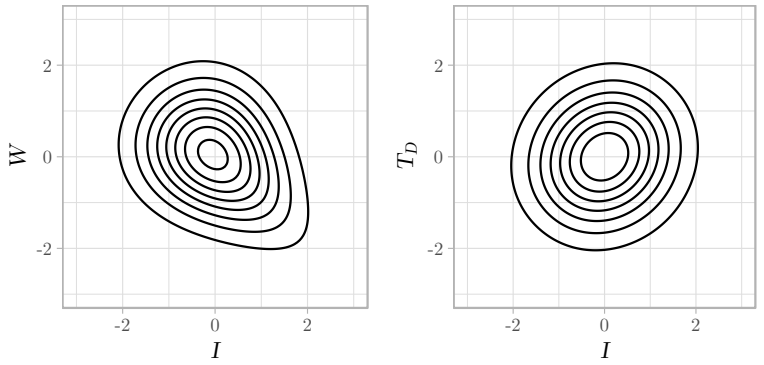
Tree	Copula	Family	θ	δ	p-value
T_1	$C_{I,W}$	Clayton 90°	0.42	–	0.82
	C_{I,T_D}	Frank	0.52	–	0.1
T_2	$C_{W,T_D\|I}$	Independence	–	–	–

Table 5.11: Full censoring of the small weather model for summer. Rotated copulae are shown by the following degree. θ denotes the first copula parameter, and δ the second.



(a) Clayton 90° **(b)** Frank **(c)** Joe 180°

Figure 5.10: Density plots of the copulae in the intensity-duration-temperature model for winter. Copula parameters are Clayton($\theta = 0.26$), Frank($\theta = 1.67$) and Joe($\theta = 1.12$). The margins are standard normal.



(a) Clayton 90° **(b)** Frank

Figure 5.11: Density plots of the copulae in the intensity-duration-temperature model for summer. Copula parameters are Clayton($\theta = 0.42$) and Frank($\theta = 0.52$). The margins are standard normal.

Chapter 6

Conclusion

6.1 Concluding Remarks

In this thesis, we have presented a multivariate model for the joint behaviour of precipitation and temperature. Existing models are mainly restricted to the bivariate case, and do not account for temperature. Furthermore, ties are also commonly present in hydrological data, but are often disregarded and rarely mentioned in the literature. In the model presented here we adapt interval censoring to common estimation and selection procedures of regular vine copulae, to account for the estimation bias induced by ties. Such construction are general in nature, and allow for joint modelling of multiple non-Gaussian stochastic variables, such as precipitation event parameters. Commonly, these do not include temperature. The events can, however, be considered a stretch from an irregular time series, which makes adding another term non-trivial. Temperature has a clear trend, which we have showed that can effectively be modelled by Fourier terms with ARIMA correction. These advancements lead to a full precipitation-temperature model that accounts for estimation biases commonly disregarded in the literature.

Two methods for estimating interval censored regular vines were proposed and tested in a large scale simulation study. The first tree is estimated equally between the methods, but they differ in the nested estimation of the marginal distributions, which is relevant for sequential trees. The first method, denoted *simple censoring*, estimates the marginal distributions by average ranks, and first considers ties when each bivariate copula is estimated. The limits for interval censoring are estimated from maximum and minimum pseudo-observations of the average rank marginal estimates. In the second method, denoted *full censoring*, the marginal distributions are approximated by both maximum and minimum ranks. In the first tree, the maximum rank observations will always be larger than the minimum rank observations, but the nested evaluation applied in higher trees will not always maintain these bounds. Hence, the censoring limits are selected as the maximum and minimum of the marginal estimates.

In the first tree, the methods show strong evidence of unbiasedness, but this does not generally hold in sequential trees, in particular for strong correlations. Still, there was an

improvement over other approaches, such as randomization and averaging. Ultimately, the method denoted *full censoring* showed the best results, which was most apparent when the key feature of the Clayton copula, the tail dependence, was censored. Some error may, however, be attributed to the vine copulae modelling approach estimated sequentially, which loses some asymptotic efficiency. Even for a sample size $n = 5000$, the untied vine estimates do not consistently reproduce the true model. In combination with severely tied data, the information loss is difficult to account for. In Haff (2012), a joint optimization of all vine parameters showed improvements to the asymptotic performance, but in this study, the optimization issues were too severe to draw the same conclusion.

The temperature and precipitation models of this study were, in part, limited by the poor data quality. Furthermore, it was apparent that the dependence between temperature and precipitation was weaker than expected in the selected region. Despite this, we have demonstrated two conceptual interval censored weather models. The first is larger and considers 6 weather parameters. Winter, spring and summer were modelled by the same vine structure, while this was different for fall. The second model investigated the relationship between intensity, duration and temperature for each season. Intensity and temperature show stronger dependence in winter, whereas intensity and duration is stronger in summer. The model is conceptual and generally applicable, so similar steps can be taken to model the dependence between temperature and precipitation in other regions.

6.2 Future Work

In regards to the precipitation events, a good start will be building on the model presented here with data of higher quality, or in a region with stronger temperature dependence. The model could then be applied to look for regional differences in dependence. Extreme compound events are in short, events where each contributing factor in it self is not extreme, but the jointly, they produce an extreme compound event. Copulae can be used to model the contributing factors, and similar to Bevacqua et al. (2017), be used to quantify risk of such events.

From a theoretical point of view, future work may include an investigation of the censored likelihood function. In particular the implications of ties in conditional distributions, and whether these can be addressed with further modifications. Under interval censoring, conditional ties are currently not explicitly expressed in the adjusted pseudo-likelihood function. The weak result of the interval censored vines in later trees is in part suspected to be a consequence of this. We showed, in Section 3.6.3, that ties in the conditional distribution *can* affect the subsequent ranks. Hence, this might be an interesting starting point.

From a more practical point of view, improvements to the optimization scheme could yield better results. The interval censored vines are currently only implemented with numerical approximations of the gradient and Hessian. This will make the joint optimization of all vine parameters a more accurate and feasible option. At this time, the joint optimization of a four dimensional vine can take up to an hour, when the data is severely tied.

References

Aas, K., Czado, C., Frigessi, A., & Bakken, H. (2009). Pair-copula constructions of multiple dependence. *Insurance: Mathematics and Economics*, *44*(2), 182 - 198. Retrieved from http://www.sciencedirect.com/science/article/pii/S0167668707000194 doi: https://doi.org/10.1016/j.insmatheco.2007.02.001

Akaike, H. (1974, December). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*(6), 716-723. doi: 10.1109/TAC.1974.1100705

Arnold, J. B. (2018). ggthemes: Extra Themes, Scales and Geoms for 'ggplot2' [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=ggthemes (R package version 4.0.1)

Aue, A., & Horváth, L. (2013). Structural breaks in time series. *Journal of Time Series Analysis*, *34*(1), 1-16. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9892.2012.00819.x doi: 10.1111/j.1467-9892.2012.00819.x

Bache, S. M., & Wickham, H. (2014). magrittr: A Forward-Pipe Operator for R [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=magrittr (R package version 1.5)

Bandeen-Roche, K. J., & Liang, K.-Y. (1996). Modelling Failure-Time Associations in Data with Multiple Levels of Clustering. *Biometrika*, *83*(1), 29–39. Retrieved from http://www.jstor.org/stable/2337430

Bedford, T., & Cooke, R. M. (2001, Aug 01). Probability Density Decomposition for Conditionally Dependent Random Variables Modeled by Vines. *Annals of Mathematics and Artificial Intelligence*, *32*(1), 245–268. Retrieved from https://doi.org/10.1023/A:1016725902970 doi: 10.1023/A:1016725902970

Bedford, T., & Cooke, R. M. (2002, 08). Vines–a new graphical model for dependent random variables. *Ann. Statist.*, *30*(4), 1031–1068. Retrieved from https://doi.org/10.1214/aos/1031689016 doi: 10.1214/aos/1031689016

Berg, D. (2009). Copula goodness-of-fit testing: an overview and power comparison. *The European Journal of Finance*, *15*(7-8), 675-701. Retrieved from https://EconPapers.repec.org/RePEc:taf:eurjfi:v:15:y:2009:i:7-8:p:675-701

Bevacqua, E., Maraun, D., Hobæk Haff, I., Widmann, M., & Vrac, M. (2017). Multivariate statistical modelling of compound events via pair-copula constructions: analysis of floods in Ravenna (Italy). *Hydrology and Earth System Sciences*, *21*(6), 2701–2723. Retrieved from https://www.hydrol-earth-syst-sci.net/21/2701/2017/ doi: 10.5194/hess-21-2701-2017

Birketvedt, J. Ø. (2019). *Bivariate Copula Analysis for High Resolution Precipitation Data.* (Project thesis). Department of Mathematical Sciences, Norwegian University of Science and Technology. (unpublished)

Brechmann, E. C. (2010). *Truncated and simplified regular vines and their applications.* (Diploma thesis). Center for Mathematical Sciences, Technische Universität München.

Brechmann, E. C., Czado, C., & Aas, K. (2012). Truncated regular vines in high dimensions with application to financial data. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, *40*(1), 68–85. Retrieved from http://www.jstor.org/stable/41724516

Brockwell, P. J., & Davis, R. A. (1987). *Time Series: Theory and Methods*. Springer-Verlag, New York.

Bücher, A., Fermanian, J.-D., & Kojadinovic, I. (2019). Combining Cumulative Sum Change-Point Detection Tests for Assessing the Stationarity of Univariate Time Series. *Journal of Time Series Analysis*, *40*(1), 124-150. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1111/jtsa.12431 doi: 10.1111/jtsa.12431

Bücher, A., & Kojadinovic, I. (2015, 12). An Overview of Nonparametric Tests of Extreme-Value Dependence and of Some Related Statistical Procedures: Methods and Applications. In (p. 377-398). doi: 10.1201/b19721-19

Bücher, A., & Kojadinovic, I. (2016, 05). A dependent multiplier bootstrap for the sequential empirical copula process under strong mixing. *Bernoulli*, *22*(2), 927–968. Retrieved from https://doi.org/10.3150/14-BEJ682 doi: 10.3150/14-BEJ682

Bücher, A., Kojadinovic, I., Rohmer, T., & Segers, J. (2014). Detecting changes in cross-sectional dependence in multivariate time series. *Journal of Multivariate Analysis*, *132*, 111 - 128. Retrieved from http://www.sciencedirect.com/science/article/pii/S0047259X14001699 doi: https://doi.org/10.1016/j.jmva.2014.07.012

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms, Third Edition* (3rd ed.). The MIT Press.

Cramér, H. (1946). *Mathematical Methods of Statistics*. Princeton University Press, Princeton.

Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal*, *Complex Systems*, 1695. Retrieved from http://igraph.org

Csörgö, M., & Horváth, L. (1997). *Limit Theorems in Change-Point Analysis*. Chichester: Wiley.

Dannevig, P. (2019, 01 28). *årstider - klima.* Retrieved 2019-05-16, from https://snl.no/%C3%A5rstider_-_klima

Deheuvels, P. (1979). La fonction de dépendence empirique et ses propriétés. Un test non paramétrique. d'indépendance. *Bulletin Royal Belge de l'Académie des Sciences*, *65*, 274–292.

Deheuvels, P. (1981, 01). A non parametric test for independence. *Publications de l'Institut de Statistique de l'Université de Paris*, *26*, 29–50.

De Michele, C., & Salvadori, G. (2003). A Generalized Pareto intensity-duration model of storm rainfall exploiting 2-Copulas. *Journal of Geophysical Research: Atmospheres*, *108*(D2). Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2002JD002534 doi: 10.1029/2002JD002534

Devroye, L. (1986). *Non-Uniform Random Variate Generation*. Springer, New York.

Dißmann, J., Christian Brechmann, E., Czado, C., & Kurowicka, D. (2012, 02). Selecting and Estimating Regular Vine Copulae and Application to Financial Returns. *Computational Statistics & Data Analysis*, *59*. doi: 10.1016/j.csda.2012.08.010

Fisher, R. (1932). *Statistical Methods for Research Workers*. Oliver and Boyd, London.

Genest, C., & Favre, A.-C. (2007). Everything You Always Wanted to Know about Copula Modeling but Were Afraid to Ask. *Journal of Hydrologic Engineering*.

Genest, C., Quessy, J.-F., & Remillard, B. (2006). Goodness-of-fit Procedures for Copula Models Based on the Probability Integral Transformation. *Scandinavian Journal of Statistics*, *33*(2), 337-366. Retrieved from https://EconPapers.repec.org/RePEc:bla:scjsta:v:33:y:2006:i:2:p:337-366

Genest, C., & Rémillard, B. (2004, Dec 01). Test of independence and randomness based on the empirical copula process. *Test*, *13*(2), 335–369. Retrieved from https://doi.org/10.1007/BF02595777 doi: 10.1007/BF02595777

Genest, C., & Rémillard, B. (2008, 12). Validity of the parametric bootstrap for goodness-of-fit testing in semiparametric models. *Ann. Inst. H. Poincaré Probab. Statist.*, *44*(6), 1096–1127. Retrieved from https://doi.org/10.1214/07-AIHP148 doi: 10.1214/07-AIHP148

Genest, C., Rémillard, B., & Beaudoin, D. (2009, April). Goodness-of-fit tests for copulas: A review and a power study. *Insurance: Mathematics and Economics*, *44*(2), 199-213. Retrieved from https://ideas.repec.org/a/eee/insuma/v44y2009i2p199-213.html

Grolemund, G., & Wickham, H. (2011). Dates and Times Made Easy with lubridate. *Journal of Statistical Software*, *40*(3), 1–25. Retrieved from http://www.jstatsoft.org/v40/i03/

Grønneberg, S., & Hjort, N. L. (2014). The Copula Information Criteria. *Scandinavian Journal of Statistics*, *41*(2), 436-459. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1111/sjos.12042 doi: 10.1111/sjos.12042

Haff, I. H. (2012). Comparison of estimators for pair-copula constructions. *Journal of Multivariate Analysis*, *110*, 91 - 105. Retrieved from http://www.sciencedirect.com/science/article/pii/S0047259X11001722 (Special Issue on Copula Modeling and Dependence) doi: https://doi.org/10.1016/j.jmva.2011.08.013

Hofert, M., Kojadinovic, I., Maechler, M., & Yan, J. (2017). copula: Multivariate Dependence with Copulas [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=copula (R package version 0.999-18)

Hofert, M., Kojadinovic, I., Maechler, M., & Yan, J. (2018). *Elements of Copula Modeling with R*. Springer Use R! Series. Retrieved from http://www.springer.com/de/book/9783319896342

Hofert, M., & Mächler, M. (2011). Nested Archimedean Copulas Meet R: The nacopula Package. *Journal of Statistical Software*, *39*(9), 1–20. Retrieved from http://www.jstatsoft.org/v39/i09/

Hofert, M., Mächler, M., & McNeil, A. J. (2012). Likelihood inference for Archimedean copulas in high dimensions under known margins. *Journal of Multivariate Analysis*, *110*, 133 - 150. Retrieved from http://www.sciencedirect.com/science/article/pii/S0047259X12000607 (Special Issue on Copula Modeling and Dependence) doi: https://doi.org/10.1016/j.jmva.2012.02.019

Holmes, M., Kojadinovic, I., & Quessy, J.-F. (2013). Nonparametric tests for change-point detection à la Gombay and Horváth. *Journal of Multivariate Analysis*, *115*, 16 - 32. Retrieved from http://www.sciencedirect.com/science/article/pii/S0047259X12002278 doi: https://doi.org/10.1016/j.jmva.2012.10.004

Huang, W., & Prokhorov, A. (2014). A goodness-of-fit test for copulas. *Econometric Reviews*, *33*(7), 751-771. Retrieved from https://doi.org/10.1080/07474938.2012.690692 doi: 10.1080/07474938.2012.690692

Joe, H. (1996). Families of m-variate distributions with given margins and $m(m-1)/2$ bivariate dependence parameters. In L. Rüschendorf, B. Schweizer, & M. D. Taylor

(Eds.), *Distributions with fixed marginals and related topics* (Vol. Volume 28, pp. 120–141). Hayward, CA: Institute of Mathematical Statistics. Retrieved from https://doi.org/10.1214/lnms/1215452614 doi: 10.1214/lnms/1215452614

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

Ko, V., Hjort, N. L., & Hobæk Haff, I. (2019). Focused information criteria for copulas. *Scandinavian Journal of Statistics*, *0*(0). Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1111/sjos.12387 doi: 10.1111/sjos.12387

Kojadinovic, I. (2017). npcp: Some Nonparametric CUSUM Tests for Change-Point Detection in Possibly Multivariate Observations [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=npcp (R package version 0.1-9)

Kojadinovic, I., & Yan, J. (2010). Modeling Multivariate Distributions with Continuous Margins Using the copula R Package. *Journal of Statistical Software*, *34*(9), 1–20. Retrieved from http://www.jstatsoft.org/v34/i09/

Kojadinovic, I., & Yan, J. (2011, Apr 01). Tests of serial independence for continuous multivariate time series based on a Möbius decomposition of the independence empirical copula process. *Annals of the Institute of Statistical Mathematics*, *63*(2), 347–373. Retrieved from https://doi.org/10.1007/s10463-009-0257-x doi: 10.1007/s10463-009-0257-x

Kojadinovic, I., Yan, J., & Holmes, M. (2011). Fast Large-Sample Goodness-of-Fit Tests For Copulas. *Statistica Sinica*, *21*(2), 841–871. Retrieved from http://www.jstor.org/stable/24309543

Kruskal, J. B. (1956). On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proceedings of the American Mathematical Society*, *7*(1), 48–50. Retrieved from http://www.jstor.org/stable/2033241

Kurowicka, D., & Cooke, R. (2006). *Uncertainty Analysis with High Dimensional Dependence Modeling*. Wiley, Chichester.

Lenderink, G., & van Meijgaard, E. (2008, 07). Increase in hourly precipitation extremes beyond expectations from temperature. *Nature Geoscience - NAT GEOSCI*, *1*, 511-514. doi: 10.1038/ngeo262

Li, Y., Li, Y., Qin, Y., & Yan, J. (2016, 12). *Copula Modeling for Data with Ties* (Tech. Rep.). arxiv:1612.06968: ArXiv E-prints.

Livera, A. M. D., Hyndman, R. J., & Snyder, R. D. (2011). Forecasting Time Series With Complex Seasonal Patterns Using Exponential Smoothing. *Journal of the American Statistical Association*, *106*(496), 1513-1527. Retrieved from https://doi.org/10.1198/jasa.2011.tm09771 doi: 10.1198/jasa.2011.tm09771

LJUNG, G. M., & BOX, G. E. P. (1978, 08). On a measure of lack of fit in time series models. *Biometrika*, *65*(2), 297-303. Retrieved from https://doi.org/10.1093/biomet/65.2.297 doi: 10.1093/biomet/65.2.297

McNeil, A. J. (2008). Sampling nested Archimedean copulas. *Journal of Statistical Computation and Simulation*, *78*(6), 567-581. Retrieved from https://doi.org/10.1080/00949650701255834 doi: 10.1080/00949650701255834

Molnar, P., Fatichi, S., Gaál, L., Szolgay, J., & Burlando, P. (2015). Storm type effects on super Clausius-Clapeyron scaling of intense rainstorm properties with air temperature. *Hydrology and Earth System Sciences*, *19*(4), 1753–1766. Retrieved from https://www.hydrol-earth-syst-sci.net/19/1753/2015/ doi: 10.5194/hess-19-1753-2015

Nelsen, R. B. (2006). *An Introduction to Copulas* (Second ed.). Springer Verlag, New York.

Norgesrekorder. (n.d.). https://www.yr.no/sted/Norge/rekorder.html. (Accessed: 2019-03-06)

Oakes, D. (1982). A Model for Association in Bivariate Survival Data. *Journal of the Royal Statistical Society. Series B (Methodological)*, *44*(3), 414–422. Retrieved from http://www.jstor.org/stable/2345500

Panthou, G., Mailhot, A., Laurence, E., & Talbot, G. (2014). Relationship between Surface Temperature and Extreme Rainfalls: A Multi-Time-Scale and Event-Based Analysis. *Journal of Hydrometeorology*, *15*(5), 1999-2011. Retrieved from https://doi.org/10.1175/JHM-D-14-0020.1 doi: 10.1175/JHM-D-14-0020.1

R Core Team. (2018). R: A Language and Environment for Statistical Computing [Computer software manual]. Vienna, Austria. Retrieved from https://www.R-project.org/

Regnværet setter nye rekorder. (2014). https://www.nrk.no/ostlandssendingen/ny-regnrekord-i-oslo-1.11801279. (Accessed: 2019-03-06)

Rota, G. C. (1964). On the Foundations of Combinatorial Theory. I. Theory of Möbius Functions. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, *2*, 340 – 368.

Salvadori, G., & Michele, C. D. (2006). Statistical characterization of temporal structure of storms. *Advances in Water Resources*, *29*(6), 827 - 842. Retrieved from http://www.sciencedirect.com/science/article/pii/S030917080500196X doi: https://doi.org/10.1016/j.advwatres.2005.07.013

Salvadori, G., & Michele, C. D. (2007). On the Use of Copulas in Hydrology: Theory and Practice. *Journal of Hydrologic Engineering*, *12*(4), 369-380. Retrieved from https://ascelibrary.org/doi/abs/10.1061/%28ASCE%291084-0699%282007%2912%3A4%28369%29 doi: 10.1061/(ASCE)1084-0699(2007)12:4(369)

Schepsmeier, U., Stoeber, J., Brechmann, E. C., Graeler, B., Nagler, T., & Erhardt, T. (2018). Vinecopula: Statistical Inference of Vine Copulas [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=VineCopula (R package version 2.1.8)

Schölzel, C., & Friederichs, P. (2008). Multivariate non-normally distributed random variables in climate research – introduction to the copula approach. *Nonlinear Processes in Geophysics*, *15*(5), 761–772. Retrieved from https://www.nonlin-processes-geophys.net/15/761/2008/ doi: 10.5194/npg-15-761-2008

Shumway, R. H., & Stoffer, D. H. (2017). *Time Series Analysis and Its Applications*. Springer International Publishing.

Sklar, A. (1959). Functions de Répartition an Dimension Set Leursmarges. *Publications de L'Institut de Statistique de L'Universite de Paris*, *8*, 229–231.

Stute, W. (1984, 05). The Oscillation Behavior of Empirical Processes: The Multivariate Case. *Ann. Probab.*, *12*(2), 361–379. Retrieved from https://doi.org/10.1214/aop/1176993295 doi: 10.1214/aop/1176993295

Tippett, L. (1931). *The Method of Statistics*. Williams and Norgate, London.

Vandenberghe, S., Verhoest, N. E. C., & De Baets, B. (2010). Fitting bivariate copulas to the dependence structure between storm characteristics: A detailed analysis based on 105 year 10 min rainfall. *Water Resources Research*, *46*(1). Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2009WR007857 doi: 10.1029/2009WR007857

White, H. (1982). Maximum Likelihood Estimation of Misspecified Models. *Econometrica*, *50*(1), 1–25. Retrieved from http://www.jstor.org/stable/1912526

Wickham, H. (2017). tidyverse: Easily Install and Load the 'tidyverse' [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=tidyverse (R package version 1.2.1)

Yan, J. (2007). Enjoy the Joy of Copulas: With a Package copula. *Journal of Statistical Software*, *21*(4), 1–21. Retrieved from http://www.jstatsoft.org/v21/i04/

Appendix A

Additional Simulations

A.1 Copula Selection

In order to verify that selection by AIC (3.11) is also efficient under interval censoring, we conduct a similar experiment to Brechmann (2010). As noted in Brechmann (2010), simulation studies for accuracy of full vine selection is difficult to construct. The vine structure is typically selected based on strength of correlation, and not data characteristics, hence, we may select a different vine than the one used for sampling. The selected copulae to each edge may also be different for the different vine structures.

In this study, we also include the AICs for average and random estimation, to see if interval censoring gives any significant benefits for selection. Here samples are collected from the Clayton, Gumbel, Frank and Joe copulae, for different levels of dependence $\tau \in \{0.25, 0.5, 0.75\}$ and sample size $n \in \{500, 1000\}$. We conduct two experiments, one where ties are symmetrically binned with $b = 15$, and one where the tails are symmetrically rounded with the severity $\lambda = 0.5$. Each experiment is conducted over $R = 1000$ trials. We also include the p-value calculated from the goodness-of-fit test in Section 3.6.2. Bootstraps are computationally costly, so p-values are calculated from only 100 bootstraps. As a selection rule, computing more bootstraps would be quite demanding for large vines. In Birketvedt (2019), inspired by Vandenberghe et al. (2010), we selected copulae by their RMSE, given by

$$\mathrm{RMSE}_C = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\big(C_\theta(u_i, v_i) - C_n(u_i, v_i)\big)^2}, \tag{A.1}$$

where C_θ is the fitted copula, so for comparison, the RMSE_C is also included in the study. For simplicity, the RMSE_C was computed from average rank pseudo-observations, and the estimated copula in each method. The selection accuracy is shown in Figures A.1, A.2, A.3 and A.4. Selection by interval censored AIC is shown to be effective in this case also, but only slightly better than the AIC estimated from average ranks. The improvement occurs for $\tau = 0.25$. It should be noted that differences may become more apparent if more

copulae were considered. The RMSE_C, however, shows to be a remarkably poor selection rule for copulae with ties.

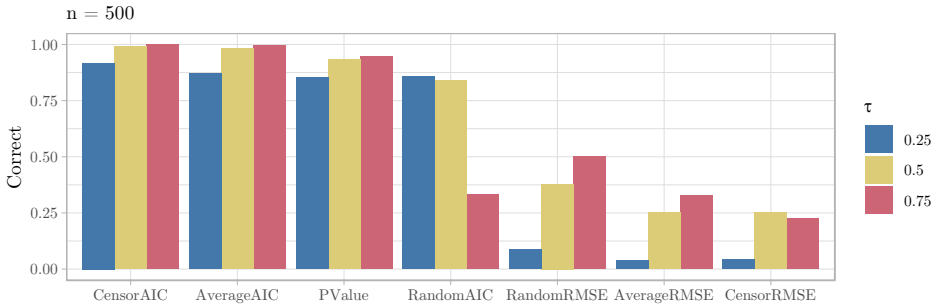


Figure A.1: Overview the amount of correct selection for each selection rule in the presence of ties. The ties are generated by symmetric binning with $b = 15$.

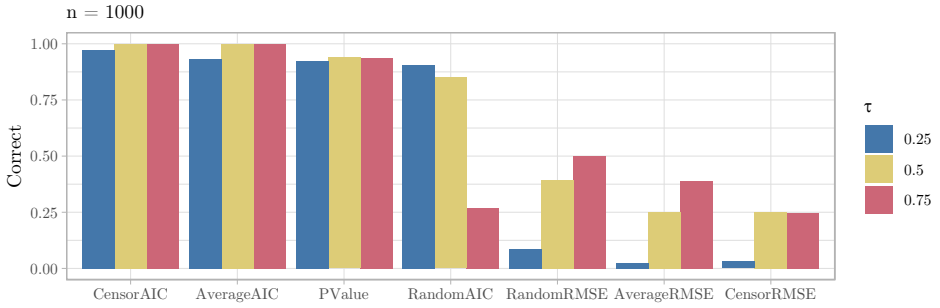


Figure A.2: Overview the amount of correct selection for each selection rule in the presence of ties. The ties are generated by symmetric binning with $b = 15$.

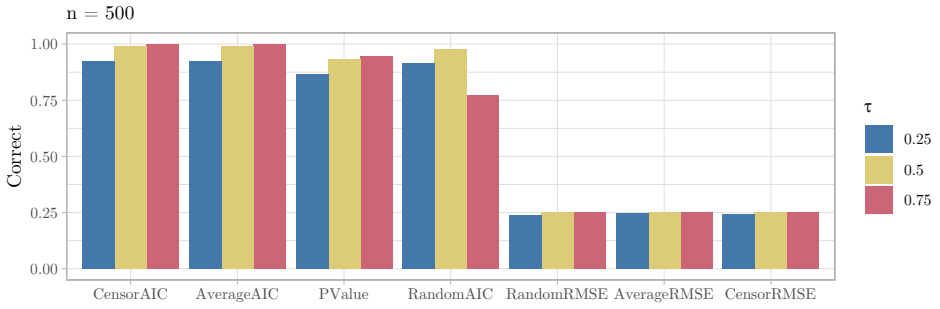


Figure A.3: Overview the amount of correct selection for each selection rule in the presence of ties. The ties are generated by symmetric tail rounding with severity $\lambda = 0.5$.

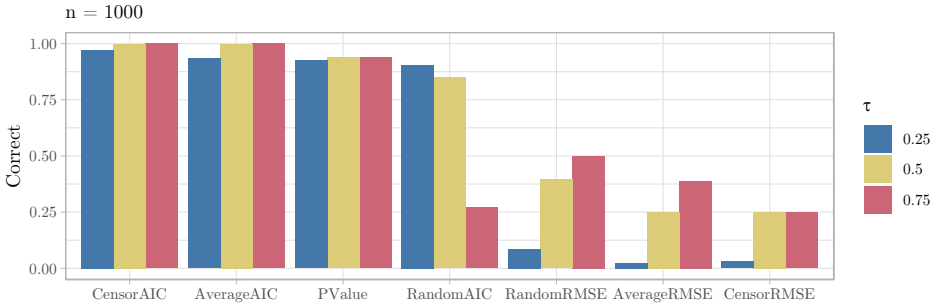


Figure A.4: Overview the amount of correct selection for each selection rule in the presence of ties. The ties are generated by symmetric tail rounding with severity $\lambda = 0.5$.

A.2 Weaker Correlation Vine

In this section we conduct a simulation study on the same vine as in Section 4.4.3, but with weaker correlations in T_2 and T_3, shown in Figure A.5. The estimation error is greatest when strong correlations are censored, so this experiment is intended to check whether the source of error in T_2, T_3 is a consequence of censored strong correlations in T_1 or strong correlations in T_2 and T_3. That is, will the estimations perform better when correlations are weaker in T_2 and T_3? We let the correlation be specified by Kendall's τ indexed by $\tau_1 \in \{0.3, 0.4, \dots 0.9\}$ for T_1, $\tau_2 \in \{0.3, 0.34, \dots, 0.56\}$ for T_2 and $\tau_3 \in \{0.2, 0.23, \dots, 37\}$ for T_3. The experiments were conducted with symmetric binning of all margins with $b = 15$ bins, see Section 4.2 for the binning procedure. Figures A.6, A.7 and A.8 show the distributions of the estimation error $\theta - \hat{\theta}$. Note that as before, the correlations in T_2 and T_3 are computed from the base correlation in T_1. They increase jointly, i.e. $\tau_1 = 0.3$ in T_1 corresponds to $\tau_2 = 0.3$ in T_2 and $\tau_3 = 0.2$ in T_3, and $\tau_1 = 0.4$ in T_1 would give $\tau_2 = 0.34$ in T_2 and $\tau_3 = 0.23$ in T_3. The correlations are still underestimated, and it appears that ties in T_1 are the primary driver for the errors.

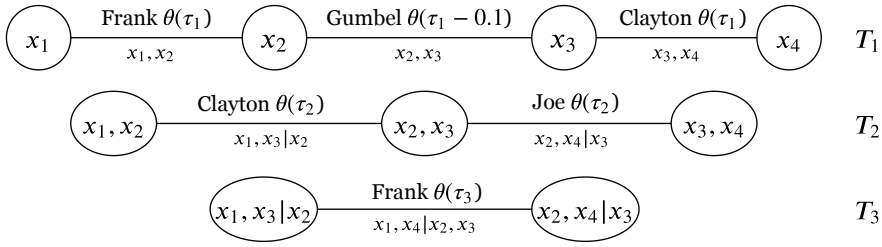


Figure A.5: Overview of the vine copula used to generate samples for the additional estimation experiment on symmetric binning. The correlations, τ_i, are used to compute the copula parameter θ for each bivariate copula. Only parameter estimates are computed for the given structure. Here the correlations are weaker in T_2 and T_3.

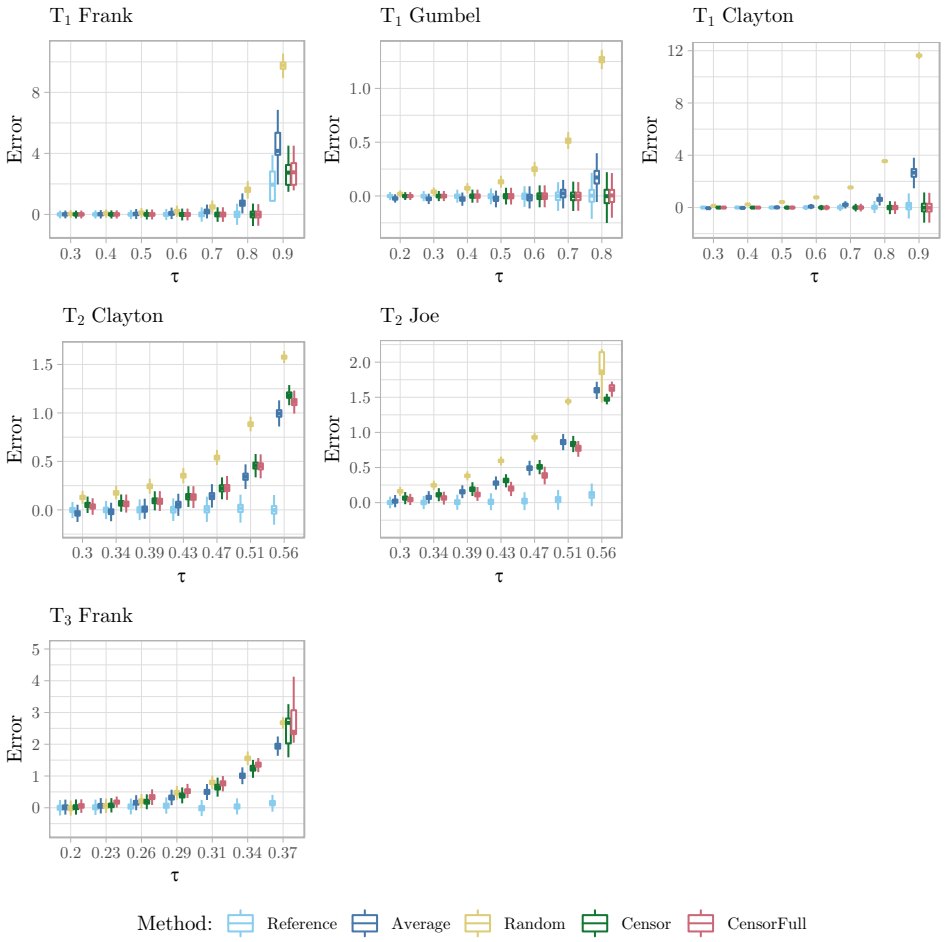


Figure A.6: Boxplot of the estimation error $\theta - \hat{\theta}$ in the symmetrically binned vine copula of Figure A.5. All four margins are binned with $b = 15$. The estimations are based on $n = 5000$ samples, and the vine parameters θ are computed from an increasing dependence τ. The correlations in T_2 and T_3 are weaker than in the experiments from Section 4.4.

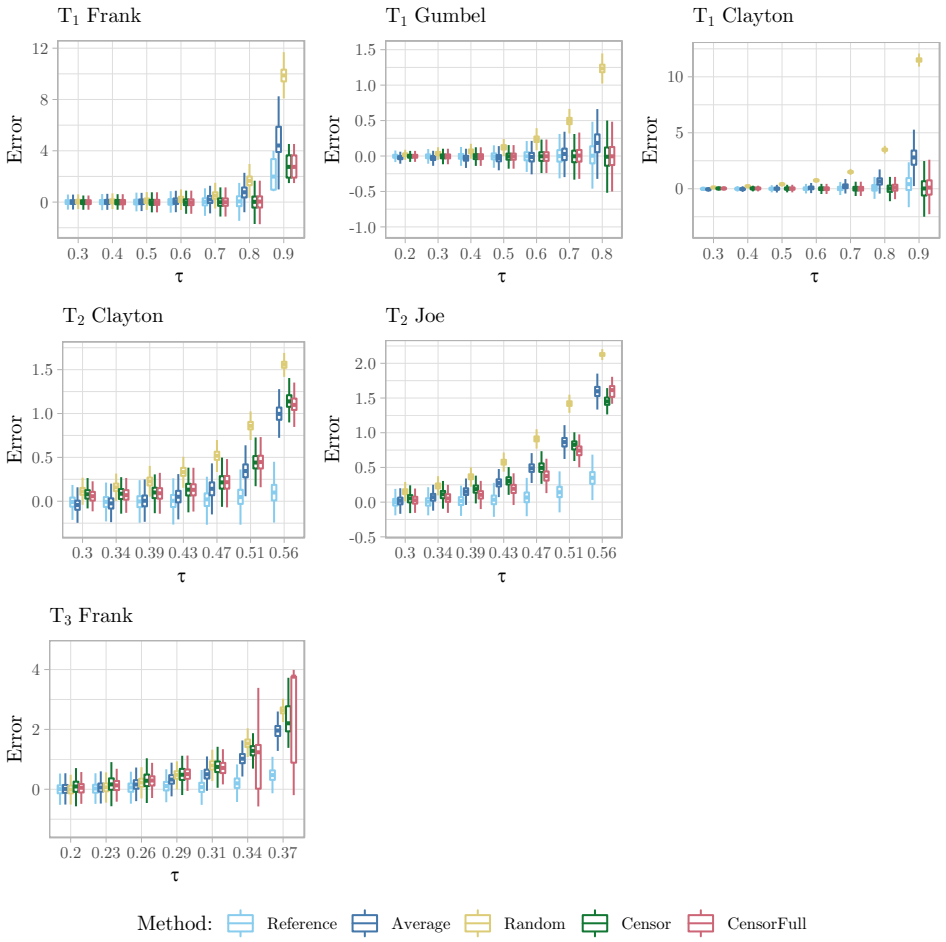


Figure A.7: Boxplot of the estimation error $\theta - \hat{\theta}$ in the symmetrically binned vine copula of Figure A.5. All four margins are binned with $b = 15$. The estimations are based on $n = 1000$ samples, and the vine parameters θ are computed from an increasing dependence τ. The correlations in T_2 and T_3 are weaker than in the experiments from Section 4.4.

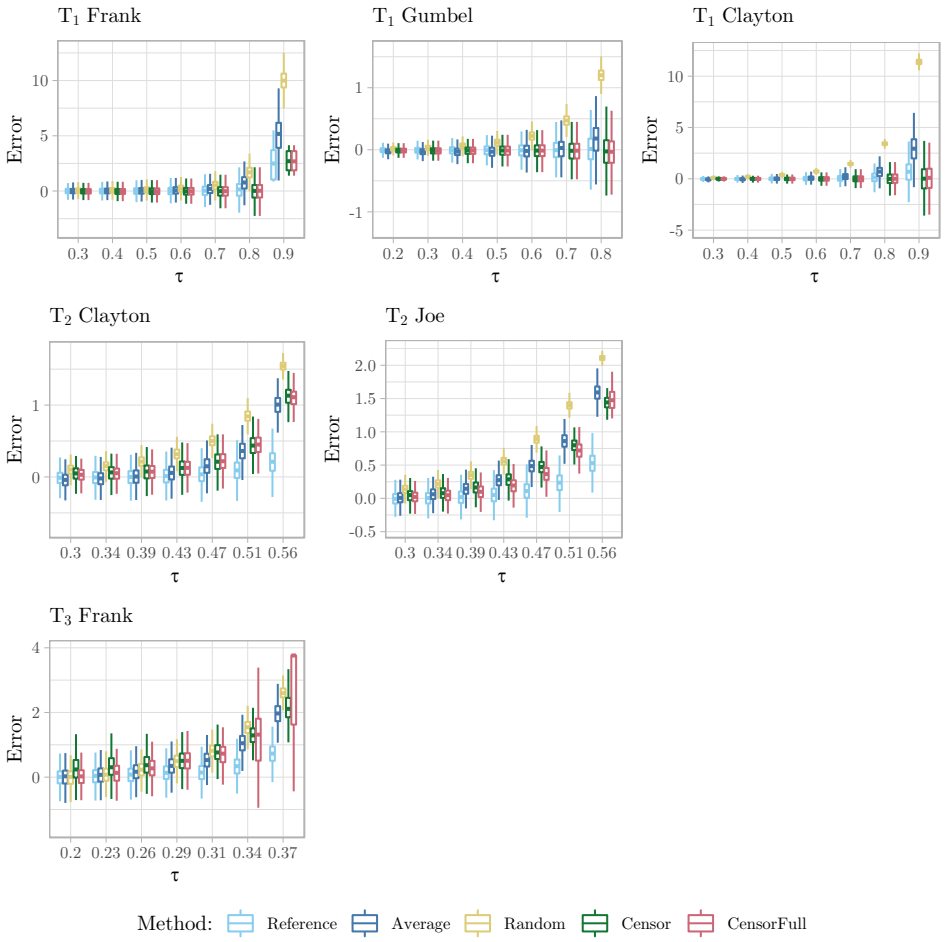


Figure A.8: Boxplot of the estimation error $\theta - \hat{\theta}$ in the symmetrically binned vine copula of Figure A.5. All four margins are binned with $b = 15$. The estimations are based on $n = 500$ samples, and the vine parameters θ are computed from an increasing dependence τ. The correlations in T_2 and T_3 are weaker than in the experiments from Section 4.4.

A.3 Additional Simulation Results

Here the remaining results from Chapter 4.4.4 are presented

A.4 Bivariate Models

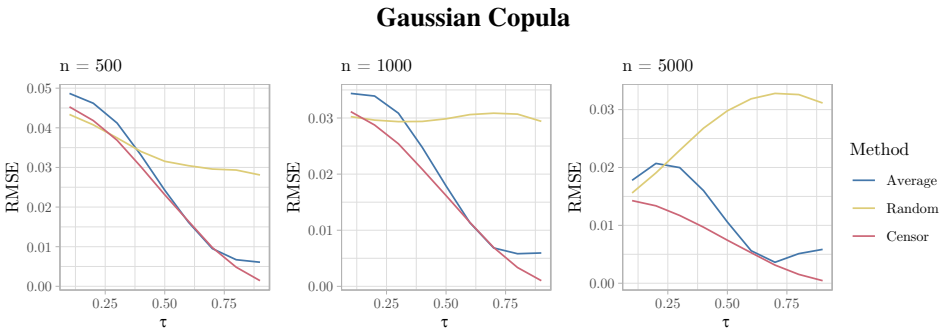


Figure A.9: RMSE in the symmetrically binned Gaussian copula. Each margin is tied in $b = 15$ bins, for a given Kendall's τ and different sample sizes n.

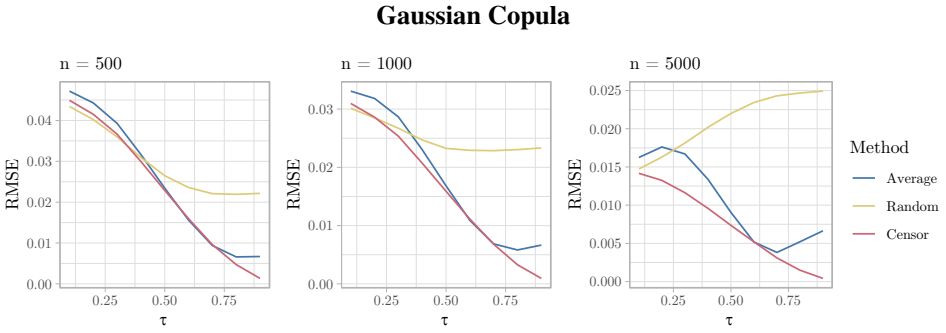


Figure A.10: RMSE in the asymmetrically binned Gaussian copula. The first margin is tied in $b_1 = 15$ bins and the second in $b_2 = 30$ bins, for a given Kendall's τ and different sample sizes n.

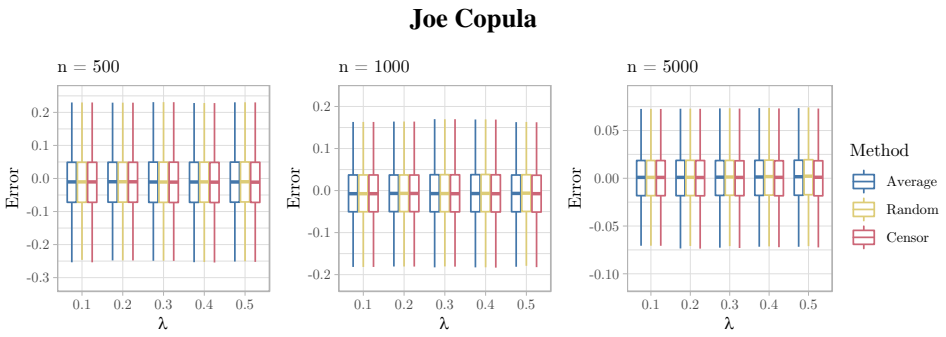


Figure A.11: Boxplot of the estimation error $\theta - \hat{\theta}$ in the Joe copula with a percentage λ of ties in the lower tails generated symmetrically. In each margin, the percentage λ of the smallest samples are rounded to the first decimal, for Kendall's $\tau = 0.25$ with increasing severity.

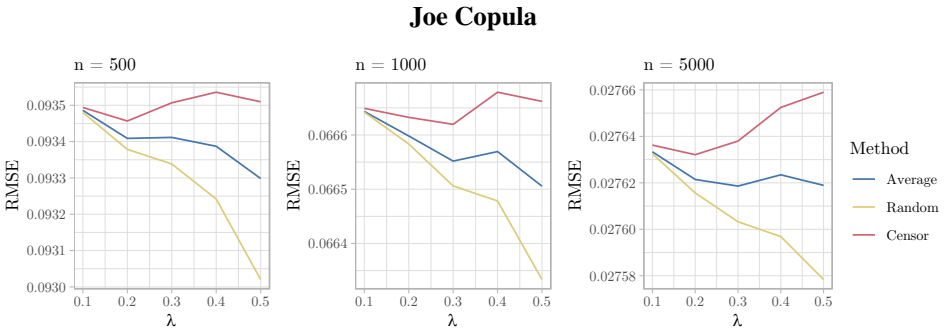


Figure A.12: RMSE in the Joe copula with a percentage λ of ties in the lower tails generated symmetrically. In each margin, the percentage λ of the smallest samples are rounded to the first decimal, for Kendall's $\tau = 0.25$ with increasing severity.

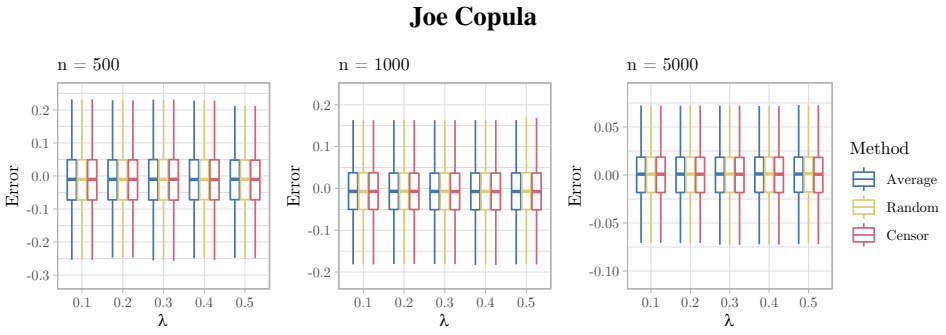


Figure A.13: Boxplot of the estimation error $\theta - \hat{\theta}$ in the Joe copula with a percentage λ of ties in the lower tails generated asymmetrically. The percentage λ of the smallest samples are rounded to the first decimal in the first margin, and to the second decimal in the second margin, for Kendall's $\tau = 0.25$ with increasing severity.

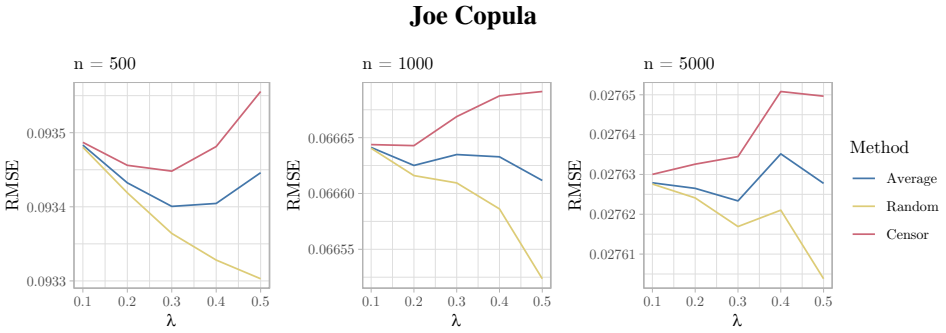


Figure A.14: RMSE in the Joe copula with a percentage λ of ties in the lower tails generated asymmetrically. The percentage λ of the smallest samples are rounded to the first decimal in the first margin, and to the second decimal in the second margin, for Kendall's $\tau = 0.25$ with increasing severity.

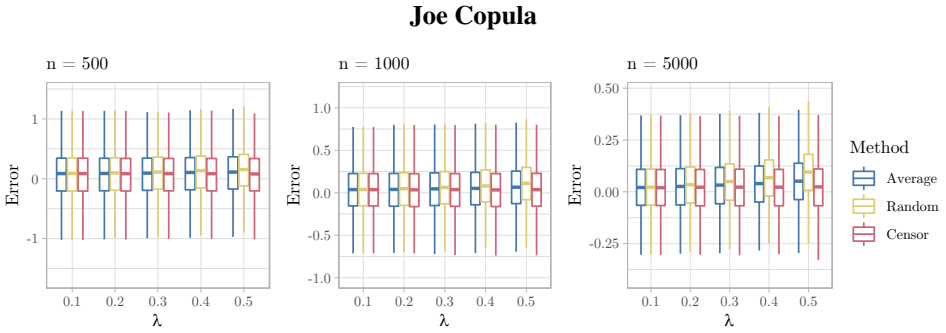


Figure A.15: Boxplot of the estimation error $\theta - \hat{\theta}$ in the Joe copula with a percentage λ of ties in the lower tails generated asymmetrically. The percentage λ of the smallest samples are rounded to the first decimal in the first margin, and to the second decimal in the second margin, for Kendall's $\tau = 0.75$ with increasing severity.

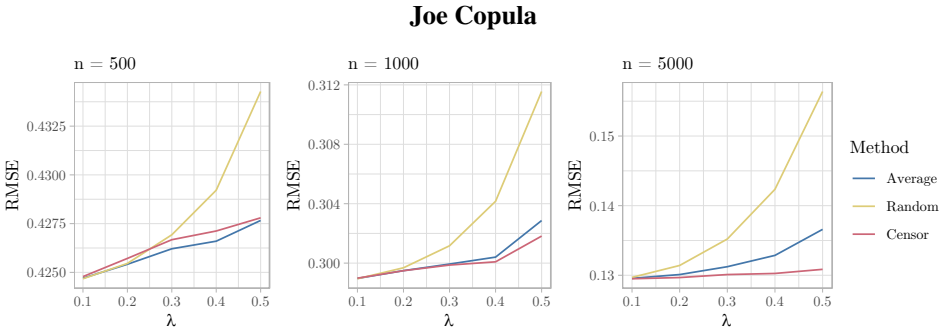


Figure A.16: RMSE in the Joe copula with a percentage λ of ties in the lower tails generated asymmetrically. The percentage λ of the smallest samples are rounded to the first decimal in the first margin, and to the second decimal in the second margin, for Kendall's $\tau = 0.75$ with increasing severity.

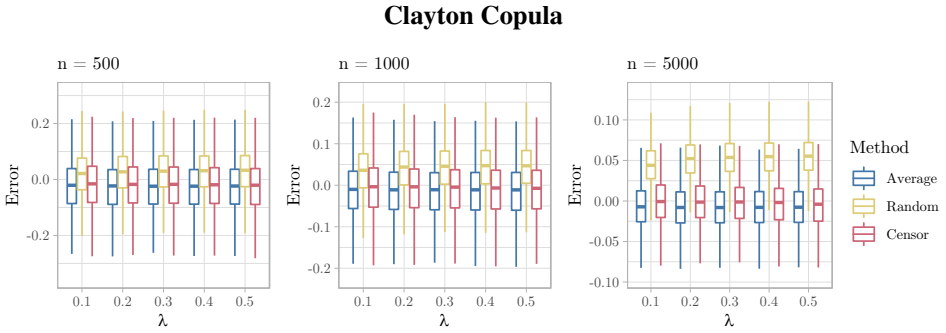


Figure A.17: Boxplot of the estimation error $\theta - \hat{\theta}$ in the Clayton copula with a percentage λ of ties in the lower tails generated asymmetrically. The percentage λ of the smallest samples are rounded to the first decimal in the first margin, and to the second decimal in the second margin, for Kendall's $\tau = 0.25$ with increasing severity.

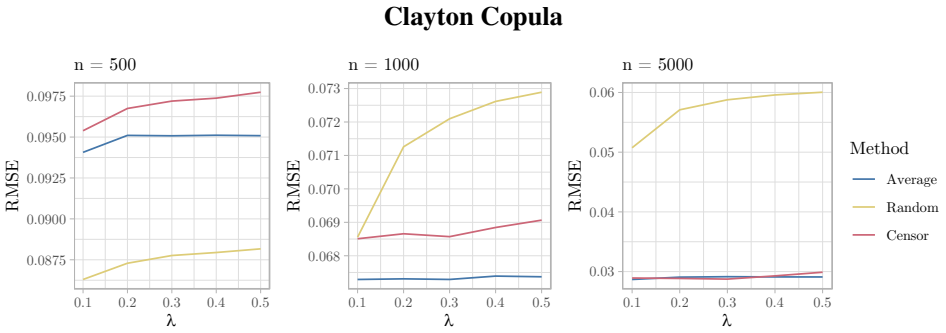


Figure A.18: RMSE in the Clayton copula with a percentage λ of ties in the lower tails generated asymmetrically. The percentage λ of the smallest samples are rounded to the first decimal in the first margin, and to the second decimal in the second margin, for Kendall's $\tau = 0.25$ with increasing severity.

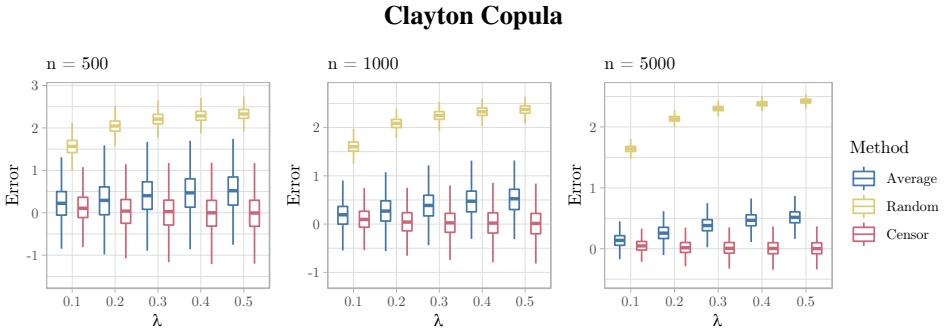


Figure A.19: Boxplot of the estimation error $\theta - \hat{\theta}$ in the Clayton copula with a percentage λ of ties in the lower tails generated symmetrically. In each margin, the percentage λ of the smallest samples are rounded to the first decimal, for Kendall's $\tau = 0.75$ with increasing severity.

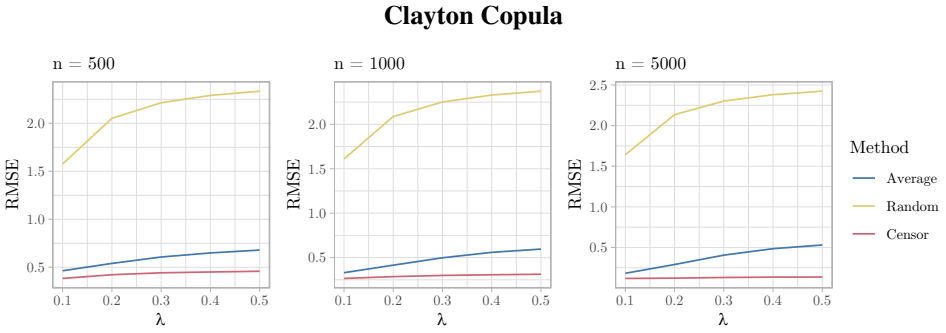


Figure A.20: RMSE in the Clayton copula with a percentage λ of ties in the lower tails generated asymmetrically. The percentage λ of the smallest samples are rounded to the first decimal in the first margin, and to the second decimal in the second margin, for Kendall's $\tau = 0.75$ with increasing severity.

A.5 Vine Models

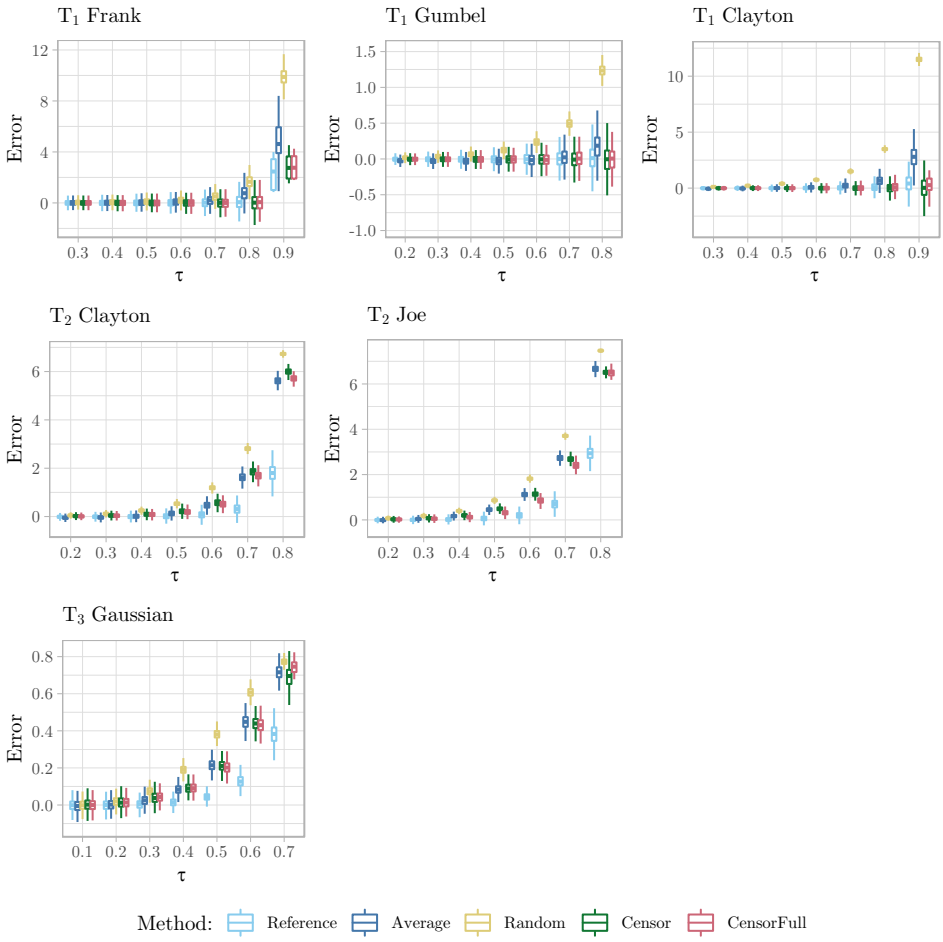


Figure A.21: Boxplot of the estimation error $\theta - \hat{\theta}$ in the symmetrically binned vine copula of Figure 4.12. All four margins are binned with $b = 15$. The estimations are based on $n = 1000$ samples, and the vine parameters θ are computed from an increasing dependence τ.

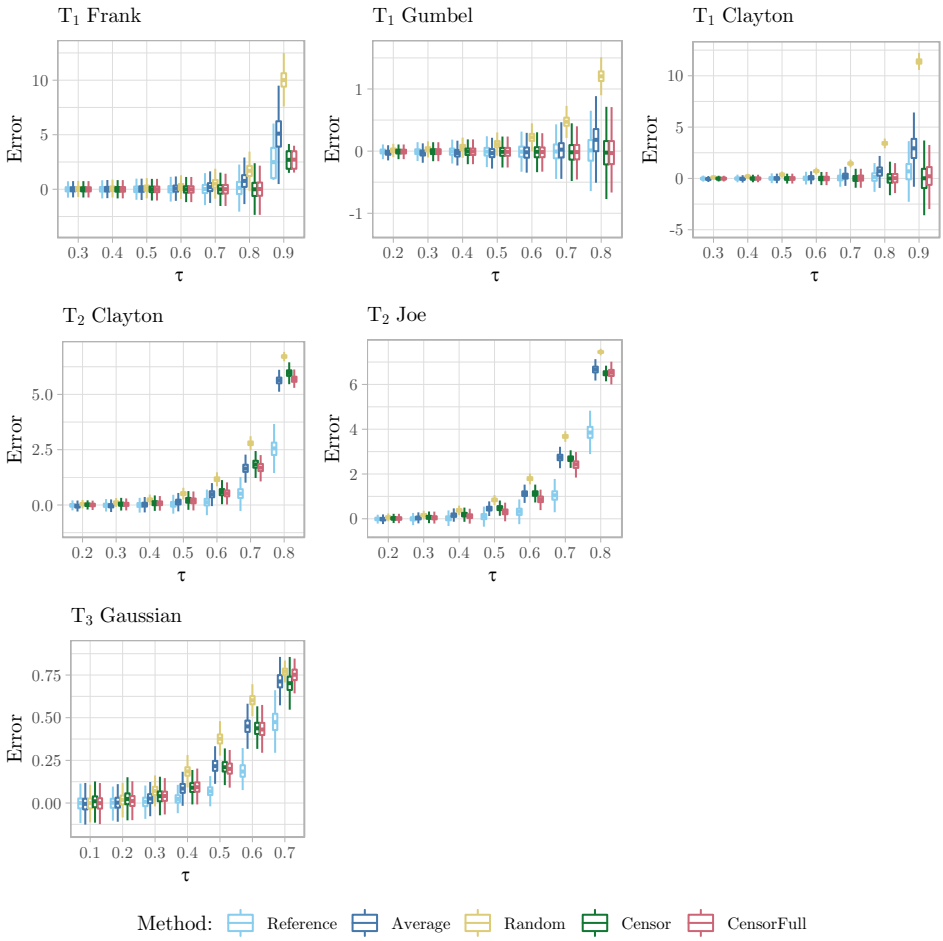


Figure A.22: Boxplot of the estimation error $\theta - \hat{\theta}$ in the symmetrically binned vine copula of Figure 4.12. All four margins are binned with $b = 15$. The estimations are based on $n = 500$ samples, and the vine parameters θ are computed from an increasing dependence τ.

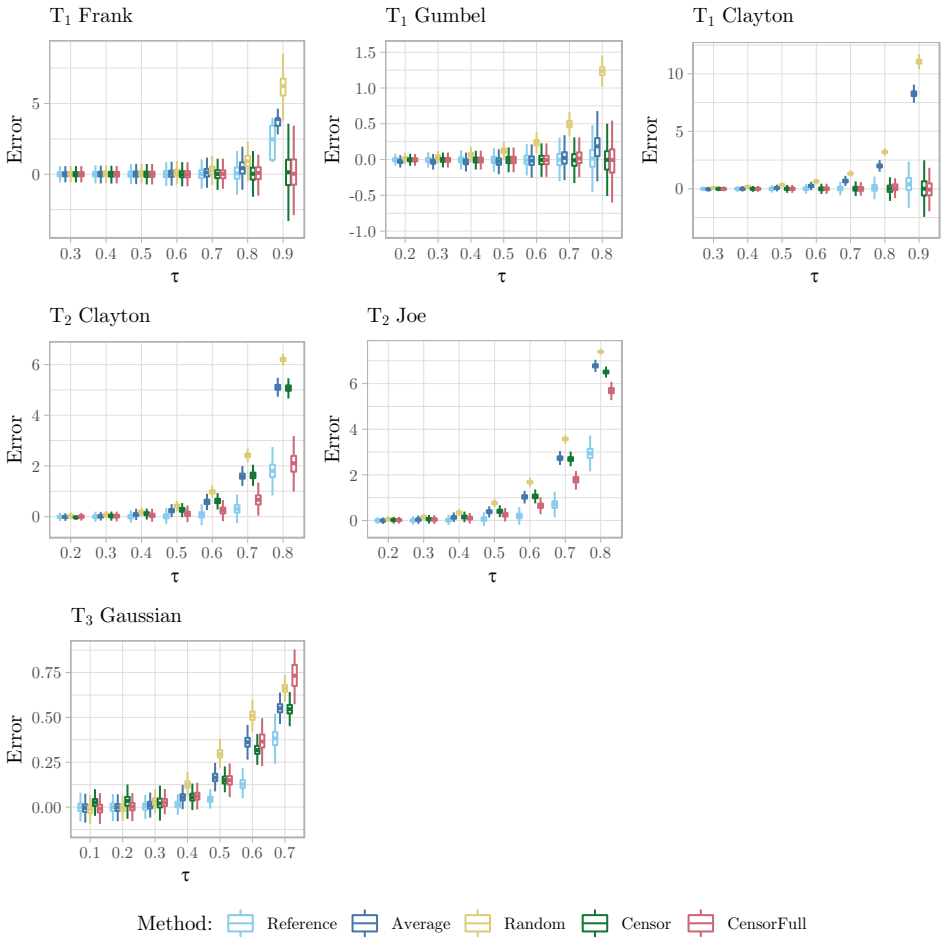


Figure A.23: Boxplot of the estimation error $\theta - \hat{\theta}$ in the asymmetrically binned vine copula of Figure 4.12. The first margin is not tied, the second and third margins are binned with $b_{2,3} = 15$, and the fourth with $b_4 = 30$. The estimations are based on $n = 1000$ samples, and the vine parameters θ are computed from an increasing dependence τ.

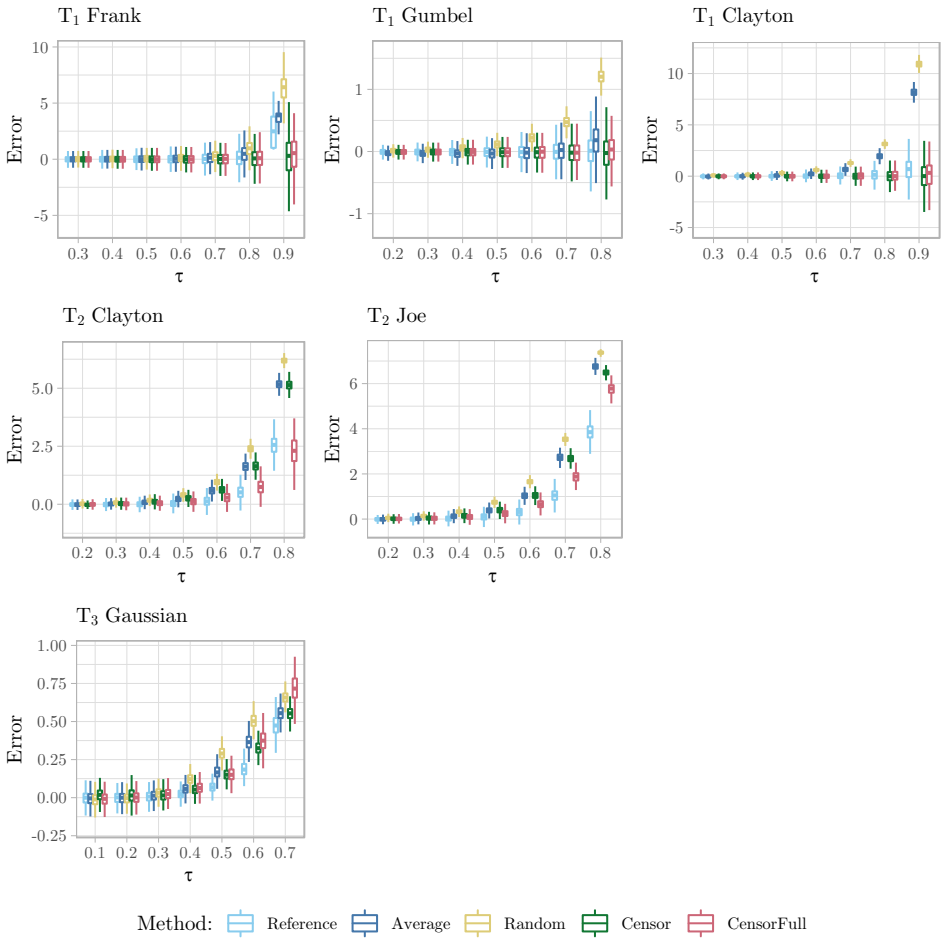


Figure A.24: Boxplot of the estimation error $\theta - \hat{\theta}$ in the asymmetrically binned vine copula of Figure 4.12. The first margin is not tied, the second and third margins are binned with $b_{2,3} = 15$, and the fourth with $b_4 = 30$. The estimations are based on $n = 500$ samples, and the vine parameters θ are computed from an increasing dependence τ.

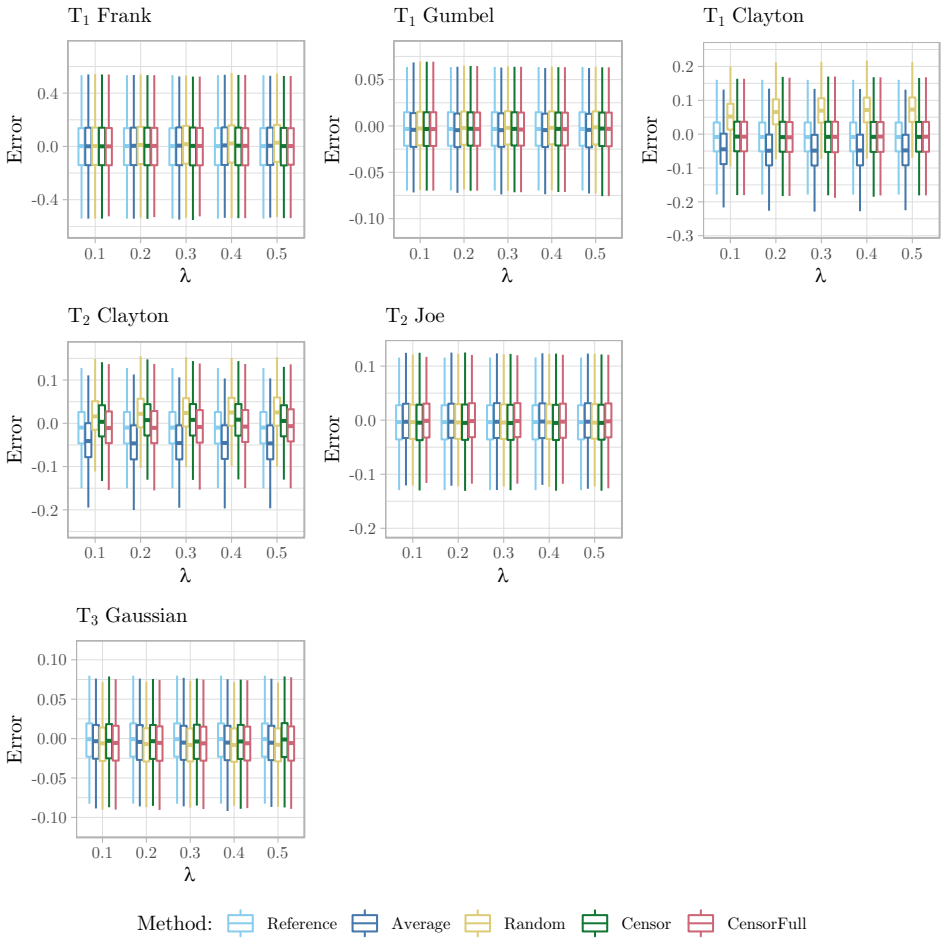


Figure A.25: Boxplot of the estimation error $\theta - \hat{\theta}$ in the vine copula 4.12 with ties generated symmetrically in the lower tails. In each margin, the percentage λ of the smallest samples are rounded to the first decimal with increasing severity. The estimations are based on $n = 1000$ samples, and the vine parameters θ are computed from a base dependence $\tau = 0.25$.

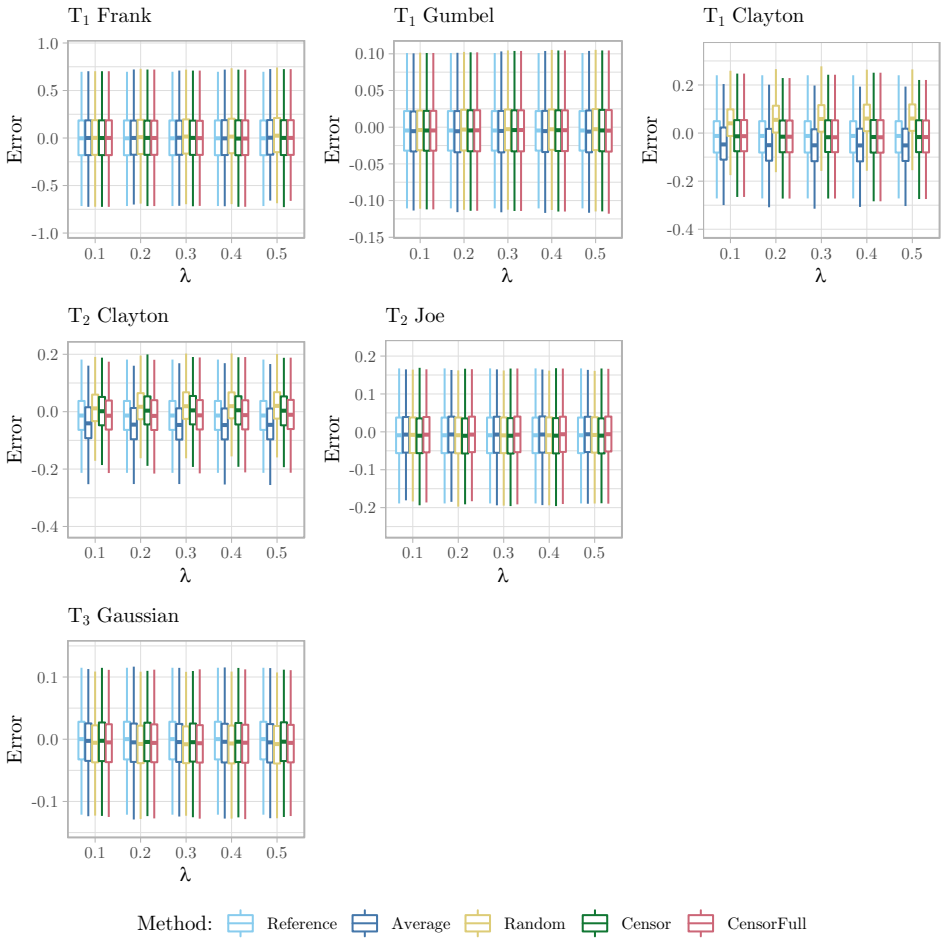


Figure A.26: Boxplot of the estimation error $\theta - \hat{\theta}$ in the vine copula 4.12 with ties generated symmetrically in the lower tails. In each margin, the percentage λ of the smallest samples are rounded to the first decimal with increasing severity. The estimations are based on $n = 500$ samples, and the vine parameters θ are computed from a base dependence $\tau = 0.25$.

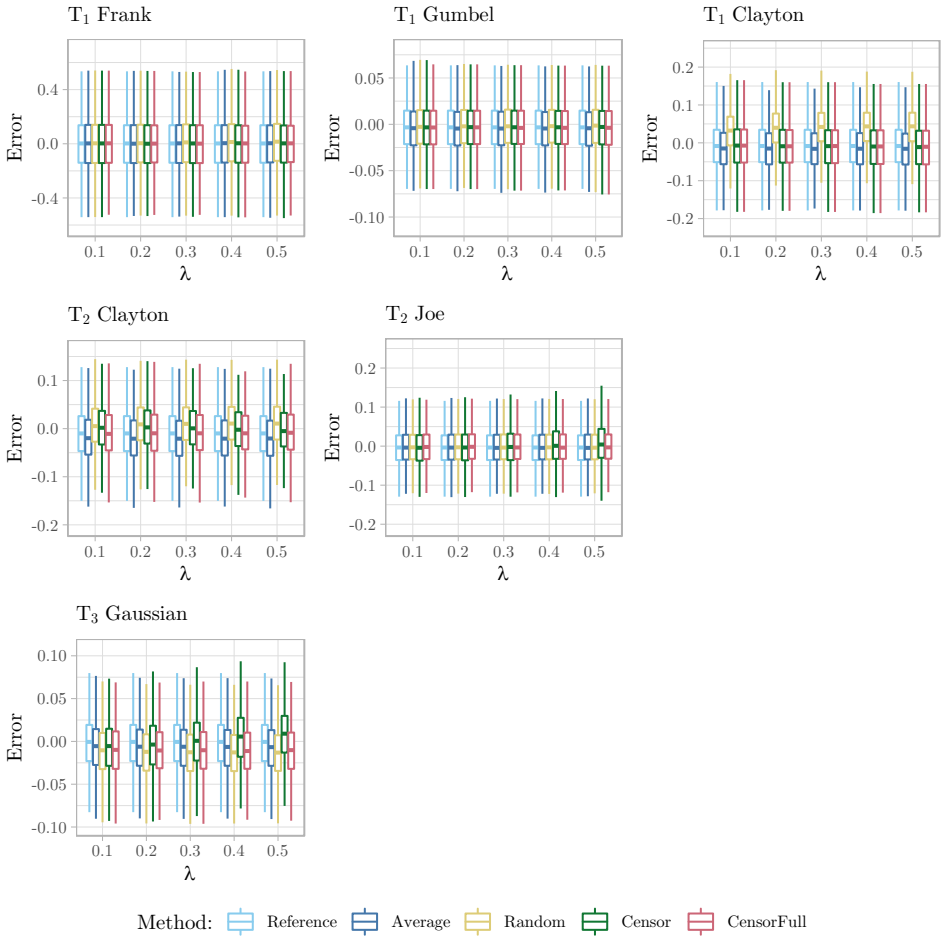


Figure A.27: Boxplot of the estimation error $\theta - \hat{\theta}$ in the vine copula 4.12 with ties generated asymmetrically in the lower tails. The percentage λ of the smallest samples are rounded to the first decimal place in the second and third margin, and to the second decimal place in the fourth margin. The first margin is not rounded. Estimations are based on $n = 1000$ samples, and the vine parameters θ are computed from a base dependence $\tau = 0.25$.

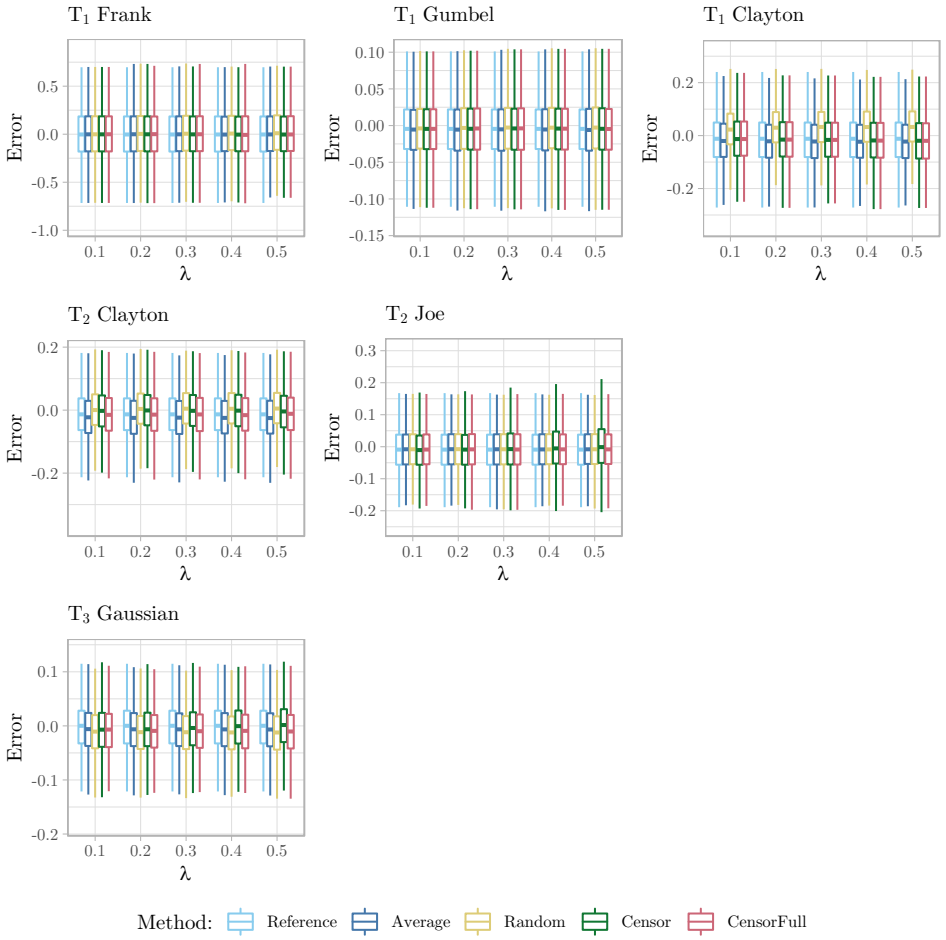


Figure A.28: Boxplot of the estimation error $\theta - \hat{\theta}$ in the vine copula 4.12 with ties generated asymmetrically in the lower tails. The percentage λ of the smallest samples are rounded to the first decimal place in the second and third margin, and to the second decimal place in the fourth margin. The first margin is not rounded. Estimations are based on $n = 500$ samples, and the vine parameters θ are computed from a base dependence $\tau = 0.25$.

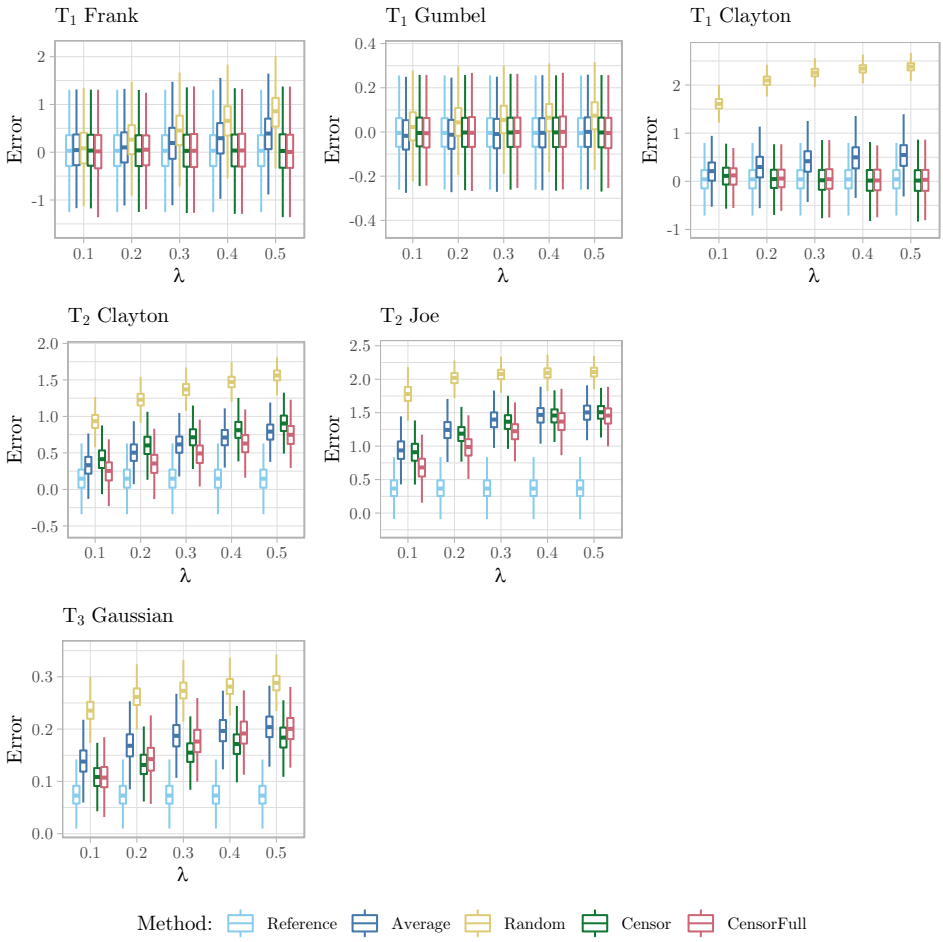


Figure A.29: Boxplot of the estimation error $\theta - \hat{\theta}$ in the vine copula 4.12 with ties generated symmetrically in the lower tails. In each margin, the percentage λ of the smallest samples are rounded to the first decimal with increasing severity. The estimations are based on $n = 1000$ samples, and the vine parameters θ are computed from a base dependence $\tau = 0.75$.

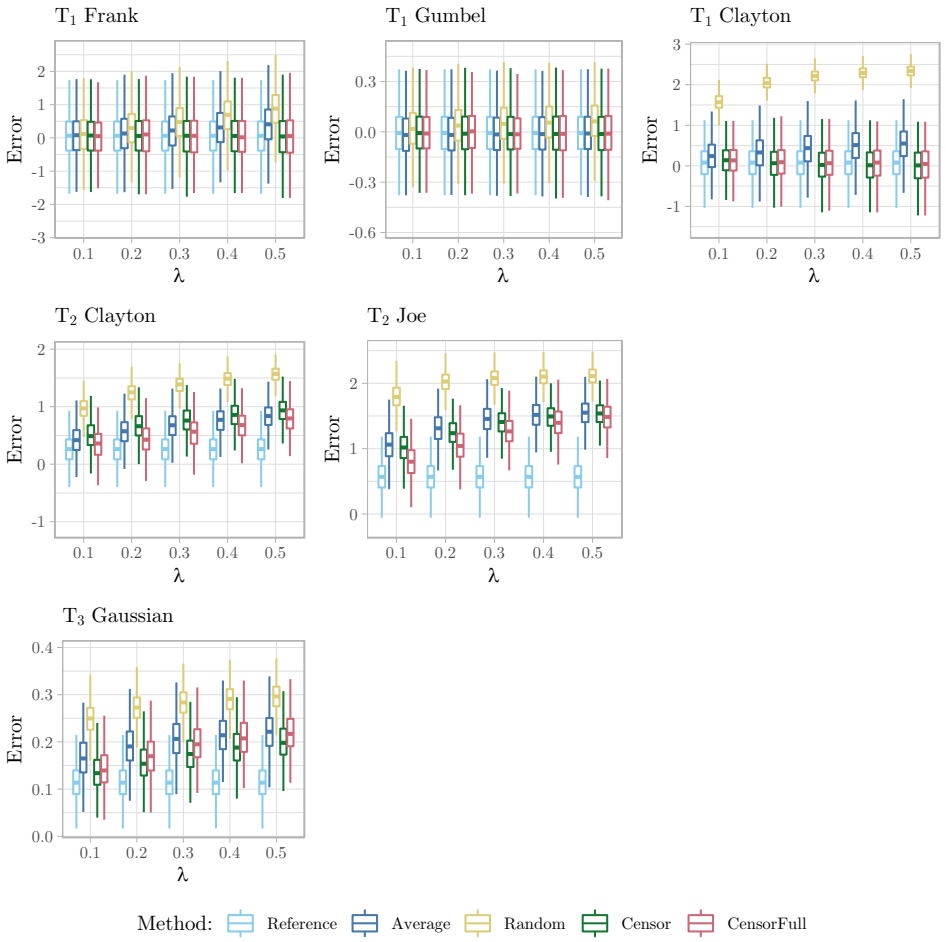


Figure A.30: Boxplot of the estimation error $\theta - \hat{\theta}$ in the vine copula 4.12 with ties generated symmetrically in the lower tails. In each margin, the percentage λ of the smallest samples are rounded to the first decimal with increasing severity. The estimations are based on $n = 500$ samples, and the vine parameters θ are computed from a base dependence $\tau = 0.75$.

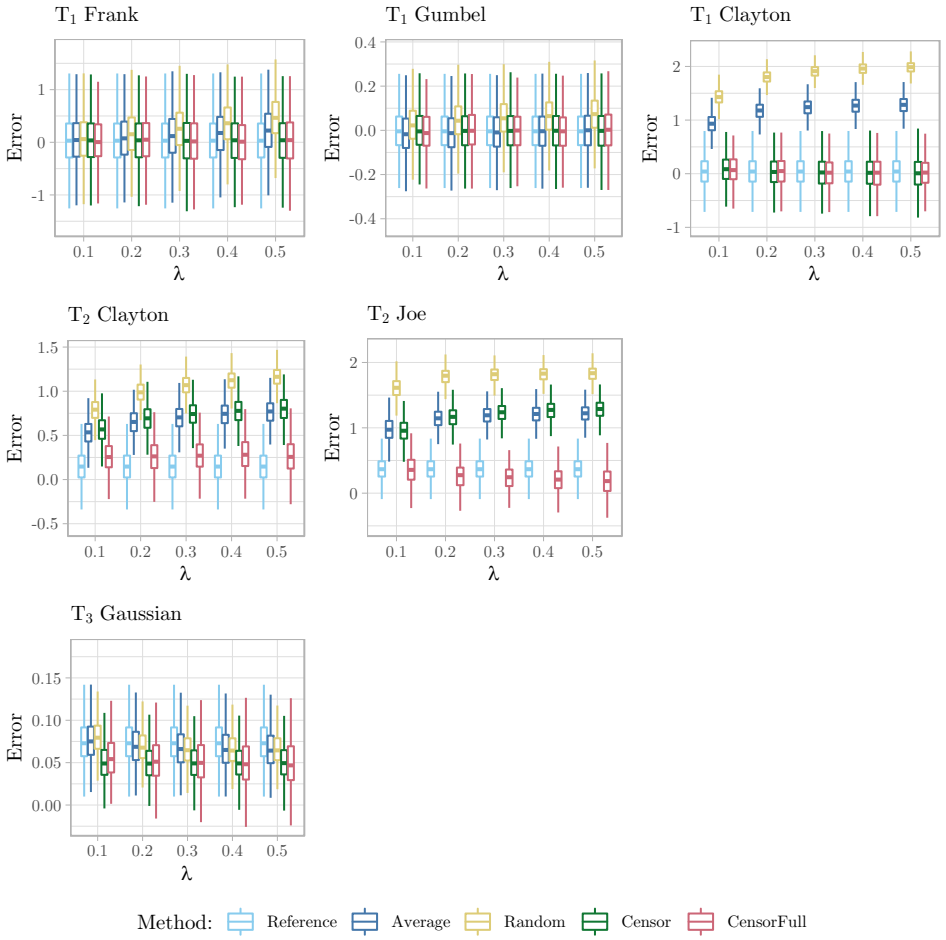


Figure A.31: Boxplot of the estimation error $\theta - \hat{\theta}$ in the vine copula 4.12 with ties generated asymmetrically in the lower tails. The percentage λ of the smallest samples are rounded to the first decimal place in the second and third margin, and to the second decimal place in the fourth margin. The first margin is not rounded. Estimations are based on $n = 1000$ samples, and the vine parameters θ are computed from a base dependence $\tau = 0.75$.

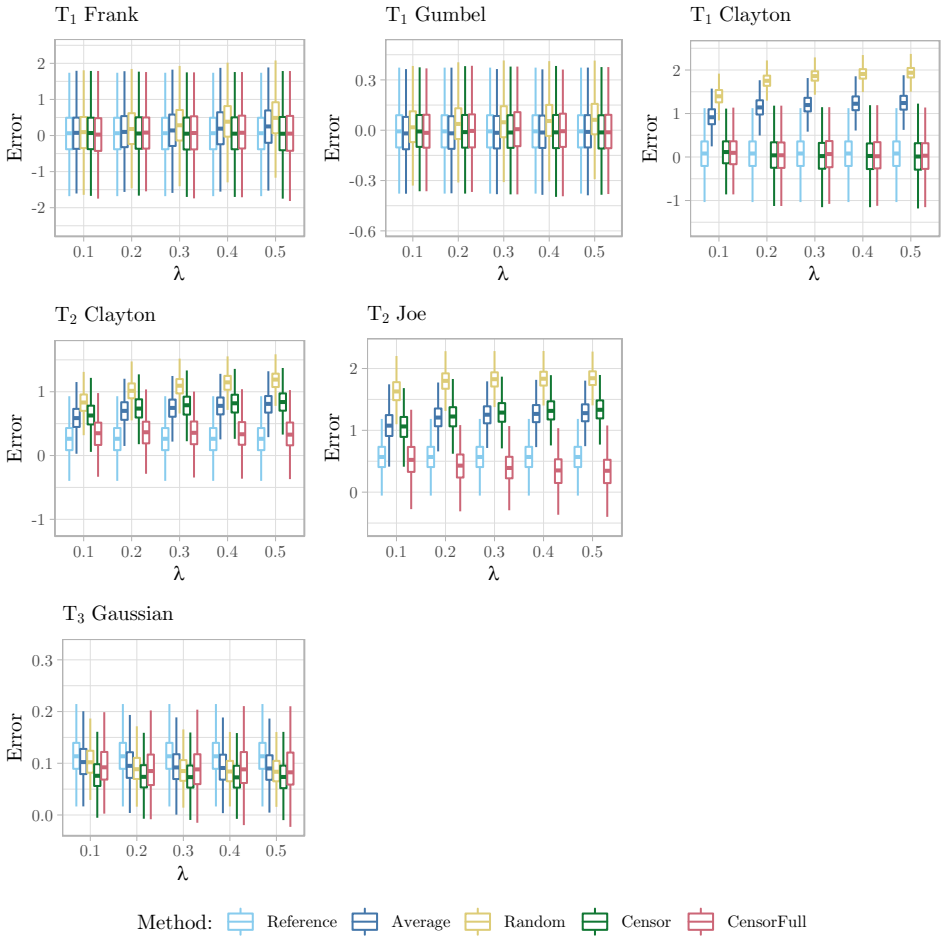


Figure A.32: Boxplot of the estimation error $\theta - \hat{\theta}$ in the vine copula 4.12 with ties generated asymmetrically in the lower tails. The percentage λ of the smallest samples are rounded to the first decimal place in the second and third margin, and to the second decimal place in the fourth margin. The first margin is not rounded. Estimations are based on $n = 500$ samples, and the vine parameters θ are computed from a base dependence $\tau = 0.75$.

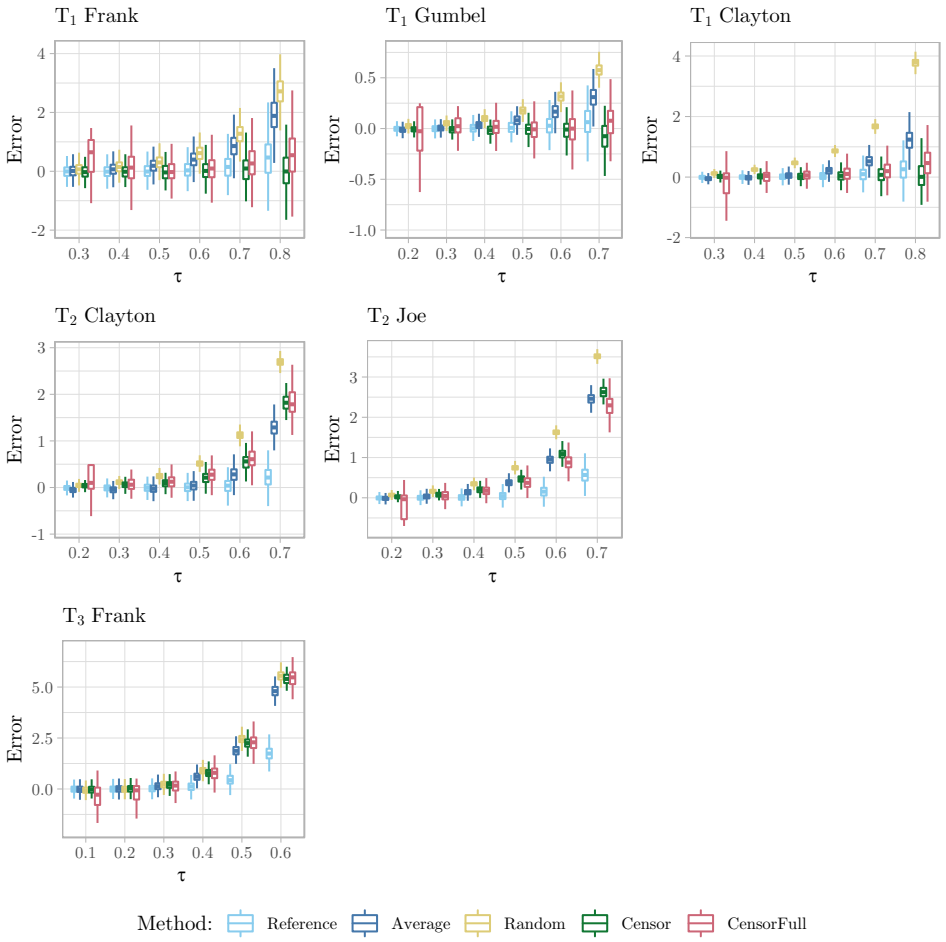


Figure A.33: Boxplot of the estimation error $\theta - \hat{\theta}$ in the symmetrically binned vine copula of Figure 4.19. All four margins are binned with $b = 15$. The estimations are based on $n = 1000$ samples, and the vine parameters θ are computed from an increasing dependence τ. The estimation is first performed sequentially, and then jointly over all vine parameters.

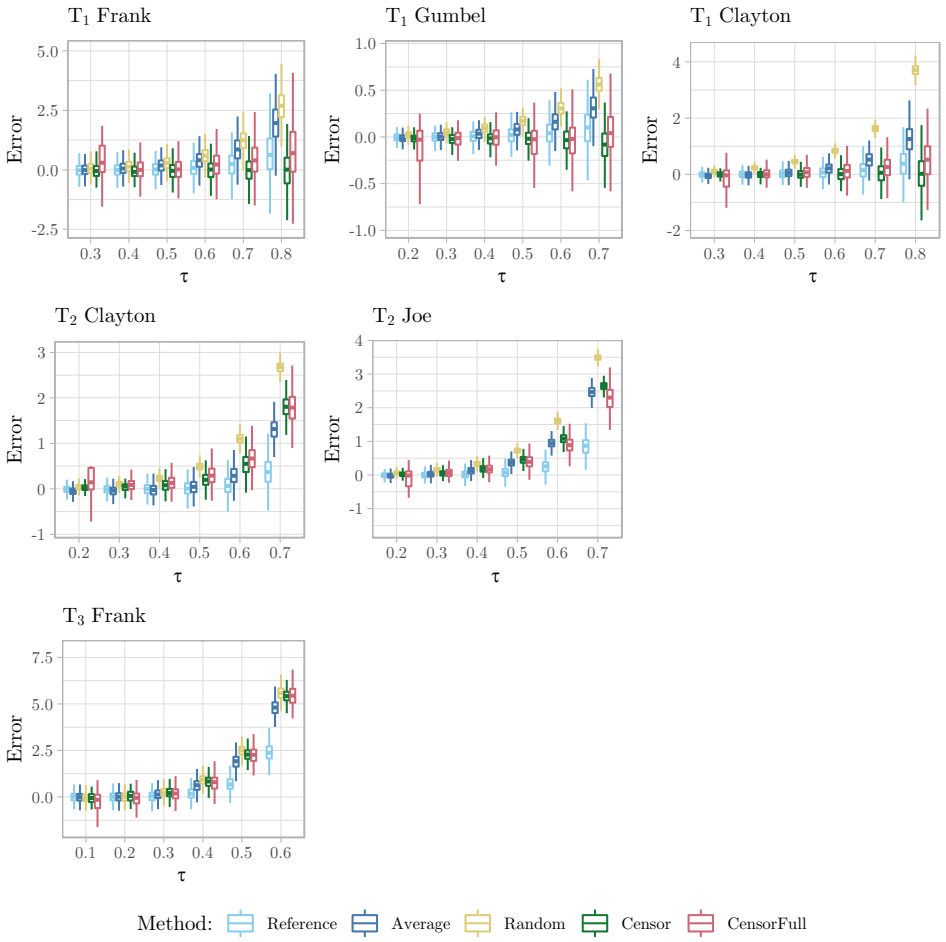


Figure A.34: Boxplot of the estimation error $\theta - \hat{\theta}$ in the symmetrically binned vine copula of Figure 4.19. All four margins are binned with $b = 15$. The estimations are based on $n = 500$ samples, and the vine parameters θ are computed from an increasing dependence τ. The estimation is first performed sequentially, and then jointly over all vine parameters.

Appendix B

Additional Copula Models for Precipitation and Temperature Data

Here we present the remaining models from Chapter 4.4.4. These are the full weather model $(I, W, V, D, T_\Delta, T_{D\Delta})$ and the smaller intensity-duration-temperature model. Also provided is the selected copula for (I, W) in fall, since this relationship was not explicitly modelled in the vine, see Table B.1.

Copula	Family	θ	δ	p-value
$C_{I,W}$	Clayton 90°	0.273	–	0.98

Table B.1: Selected copula for (I, W) in fall. Provided as an addition, since this relationship was not explicitly modelled in the vine.

B.1 Full Censoring

Tree	Copula	Family	θ	δ	p-value
T_1	$C_{V,I}$	Frank	4.44	–	0.00
	$C_{W,V}$	Gumbel	1.91	–	0.62
	$C_{T_\Delta,W}$	Gaussian	0.61	–	0.05
	$C_{T_{D\Delta},D}$	Gumbel	1.54	–	1.00
	$C_{T_{D\Delta},T_\Delta}$	Gaussian	0.1	–	0.71
T_2	$C_{W,I\|V}$	Frank	−36.86	–	0.00
	$C_{T_\Delta,V\|W}$	BB8 180°	1.20	0.89	0.58
	$C_{T_{D\Delta},W\|T_\Delta}$	Independence	–	–	–
	$C_{T_\Delta,D\|T_{D\Delta}}$	Independence	–	–	–
T_3	$C_{T_\Delta,I\|W,V}$	Clayton 180°	0.1	–	0.53
	$C_{T_{D\Delta},V\|T_\Delta,W}$	Independence	–	–	–
	$C_{D,W\|T_{D\Delta},T_\Delta}$	Frank	−0.47	–	0.10

Table B.2: Full censoring of the full weather model for winter. Rotated copulae are shown by the following degree. θ denotes the first copula parameter, and δ the second. The remaining pairs are modelled by the independence copula.

Tree	Copula	Family	θ	δ	p-value
T_1	$C_{V,I}$	Frank	4.53	–	0.00
	$C_{W,V}$	Gaussian	0.57	–	0.11
	$C_{T_\Delta,W}$	Gaussian	0.57	–	0.06
	$C_{T_{D\Delta},D}$	Gaussian	0.62	–	0.02
	$C_{T_\Delta,T_{D\Delta}}$	Gumbel	1.12	–	0.92
T_2	$C_{T_\Delta,D\|T_{D\Delta}}$	Gumbel 90°	1.06	–	0.90
	$C_{W,I\|V}$	Frank	−32.93	–	0.00
	$C_{T_\Delta,V\|W}$	Independence	–	–	–
	$C_{T_{D\Delta},W\|T_\Delta}$	Independence	–	–	–
T_4	$C_{V,D\|W,T_{D\Delta},T_\Delta}$	Frank	−0.51	–	0.68
	$C_{T_{D\Delta},I\|T_\Delta,V,W}$	Independence	–	–	–

Table B.3: Full censoring of the full weather model for spring. Rotated copulae are shown by the following degree. θ denotes the first copula parameter, and δ the second. The remaining pairs are modelled by the independence copula.

Tree	Copula	Family	θ	δ	p-value
T_1	$C_{T_{D\Delta},D}$	Gaussian	0.64	–	0.00
	$C_{V,I}$	Frank	4.81	–	0.00
	$C_{W,V}$	Gumbel	1.63	–	0.99
	$C_{T_\Delta,W}$	Gaussian	0.6	–	0.08
	$C_{T_\Delta,T_{D\Delta}}$	Clayton 180°	0.24	–	0.71
T_2	$C_{T_\Delta,D\|T_{D\Delta}}$	Gaussian	−0.10	–	0.38
	$C_{W,I\|V}$	Frank	−39.60	–	0.00
	$C_{T_\Delta,V\|W}$	Gaussian	0.10	–	0.64
	$C_{T_{D\Delta},W\|T_\Delta}$	Frank	−1.09	–	0.34
T_4	$C_{V,D\|W,T_{D\Delta},T_\Delta}$	Clayton	0.1	–	0.94
	$C_{T_{D\Delta},I\|T_\Delta,V,W}$	Independence	–	–	–

Table B.4: Full censoring of the full weather model for summer. Rotated copulae are shown by the following degree. θ denotes the first copula parameter, and δ the second. The remaining pairs are modelled by the independence copula.

Tree	Copula	Family	θ	δ	p-value
T_1	$C_{I,V}$	Frank	4.34	–	0.00
	$C_{W,V}$	Gaussian	0.69	–	0.01
	C_{W,T_Δ}	Frank	5.18	–	0.27
	$C_{D,T_{D\Delta}}$	Gumbel	1.66	–	0.96
	$C_{D,W}$	Independence	–	–	–
T_2	$C_{I,W\|V}$	Gumbel 90°	12.08	–	0.01

Table B.5: Full censoring of the full weather model for fall. Rotated copulae are shown by the following degree. θ denotes the first copula parameter, and δ the second. The remaining pairs are modelled by the independence copula.

Tree	Copula	Family	θ	δ	p-value
T_1	$C_{I,W}$	Clayton 90°	0.26	–	0.99
	$C_{I,T}$	Frank	1.67	–	0.11
T_2	$C_{W,T\|I}$	Joe 180°	1.12	–	0.90

Table B.6: Full censoring of the small weather model for winter. Rotated copulae are shown by the following degree. θ denotes the first copula parameter, and δ the second.

Tree	Copula	Family	θ	δ	p-value
T_1	C_{W,T_m}	Gaussian	−0.29	–	0.29
	$C_{I,W}$	Clayton 90°	0.45	–	0.67
T_2	$C_{I,T_m\|W}$	Gumbel 180°	1.08	–	1.00

Table B.7: Full censoring of the small weather model for spring. Rotated copulae are shown by the following degree. θ denotes the first copula parameter, and δ the second.

Tree	Copula	Family	θ	δ	p-value
T_1	$C_{I,W}$	Clayton 90°	0.42	–	0.82
	C_{I,T_D}	Frank	0.52	–	0.1
T_2	$C_{W,T_D\|I}$	Independence	–	–	–

Table B.8: Full censoring of the small weather model for summer. Rotated copulae are shown by the following degree. θ denotes the first copula parameter, and δ the second.

Tree	Copula	Family	θ	δ	p-value
T_1	$C_{T_M,W}$	Frank	1.47	–	0.02
	$C_{T_M,I}$	Gumbel	1.15	–	0.98
T_2	$C_{I,W\|T_M}$	Clayton 270°	0.36	–	0.00

Table B.9: Full censoring of the small weather model for fall. Rotated copulae are shown by the following degree. θ denotes the first copula parameter, and δ the second.

B.2 Simple Censoring

Tree	Copula	Family	θ	δ	p-value
T_1	$C_{I,V}$	Frank	4.44	–	0.00
	$C_{W,V}$	Gumbel	1.91	–	0.29
	$C_{T_\Delta,W}$	Gaussian	0.61	–	0.06
	$C_{T_{D\Delta},D}$	Gumbel	1.54	–	0.02
	$C_{T_\Delta,T_{D\Delta}}$	Gaussian	0.1	–	0.71
T_2	$C_{I,W\|V}$	Frank	-40	–	0.00
	$C_{T_\Delta,V\|W}$	Clayton	0.10	–	0.50
	$C_{T_{D\Delta},W\|T_\Delta}$	Independence	–	–	–
	$C_{T_\Delta,D\|T_{D\Delta}}$	Independence	–	–	–
T_3	$C_{T_\Delta,I\|W,V}$	Frank	0.41	–	0.78
	$C_{T_{D\Delta},V\|T_\Delta,W}$	Independence	–	–	–
	$C_{D,W\|T_{D\Delta},T_\Delta}$	Frank	-0.47	–	0.15

Table B.10: Simple censoring of the full weather model for winter. Rotated copulae are shown by the following degree. θ denotes the first copula parameter, and δ the second. The remaining pairs are modelled by the independence copula.

Tree	Copula	Family	θ	δ	p-value
T_1	$C_{I,V}$	Frank	4.53	–	0.00
	$C_{W,V}$	Gaussian	0.57	–	0.10
	$C_{T_\Delta,W}$	Gaussian	0.57	–	0.06
	$C_{T_{D\Delta},D}$	Gaussian	0.62	–	0.01
	$C_{T_\Delta,T_{D\Delta}}$	Gumbel	1.12	–	0.92
T_2	$C_{I,W\|V}$	Frank	-35.61	–	0.00
	$C_{T_\Delta,V\|W}$	Independence	–	–	–
	$C_{T_{D\Delta},W\|T_\Delta}$	Independence	–	–	–
	$C_{T_\Delta,D\|T_{D\Delta}}$	Joe 90°	1.09	–	0.96
T_4	$C_{V,D\|W,T_{D\Delta},T_\Delta}$	Gumbel	1.05	–	0.49
	$C_{T_{D\Delta},I\|T_\Delta,V,W}$	Independence	–	–	–

Table B.11: Simple censoring of the full weather model for spring. Rotated copulae are shown by the following degree. θ denotes the first copula parameter, and δ the second. The remaining pairs are modelled by the independence copula.

Tree	Copula	Family	θ	δ	p-value
T_1	$C_{I,V}$	Frank	4.81	–	0.00
	$C_{W,V}$	Gumbel	1.63	–	0.06
	$C_{T_\Delta,W}$	Gaussian	0.6	–	0.08
	$C_{T_{D\Delta},D}$	Gaussian	0.64	–	0.00
	$C_{T_\Delta,T_{D\Delta}}$	Clayton 180°	0.24	–	0.72
T_2	$C_{I,W\|V}$	Frank	-40	–	0.00
	$C_{T_\Delta,V\|W}$	Gaussian	0.10	–	0.61
	$C_{T_{D\Delta},W\|T_\Delta}$	Frank	-1.09	–	0.30
	$C_{T_\Delta,D\|T_{D\Delta}}$	Gaussian	-0.10	–	0.62
T_4	$C_{V,D\|W,T_{D\Delta},T_\Delta}$	Clayton	0.09	–	0.70
	$C_{T_{D\Delta},I\|T_\Delta,V,W}$	Independence	–	–	–

Table B.12: Simple censoring of the full weather model for summer. Rotated copulae are shown by the following degree. θ denotes the first copula parameter, and δ the second. The remaining pairs are modelled by the independence copula.

Tree	Copula	Family	θ	δ	p-value
T_1	$C_{I,V}$	Frank	4.34	–	0.00
	$C_{W,V}$	Gaussian	0.69	–	0.01
	C_{W,T_Δ}	Frank	5.18	–	0.27
	$C_{D,T_{D\Delta}}$	Gumbel	1.66	–	0.03
	$C_{D,W}$	Independence	–	–	–
T_2	$C_{I,W\|V}$	Gumbel 90°	11.84	–	0.01

Table B.13: Simple censoring of the full weather model for fall. Rotated copulae are shown by the following degree. θ denotes the first copula parameter, and δ the second. The remaining pairs are modelled by the independence copula.

Tree	Copula	Family	θ	δ	p-value
T_1	$C_{I,T}$	Frank	1.67	–	0.116
	$C_{I,W}$	Clayton 90°	0.26	–	0.99
T_2	$C_{W,T\|I}$	Joe 180°	1.10	–	0.28

Table B.14: Simple censoring of the small weather model for winter. Rotated copulae are shown by the following degree. θ denotes the first copula parameter, and δ the second.

Tree	Copula	Family	θ	δ	p-value
T_1	C_{W,T_m}	Gaussian	-0.29	–	0.29
	$C_{I,W}$	Clayton 90°	0.45	–	0.67
T_2	$C_{I,T_m\|W}$	Gumbel 180°	1.07	–	0.88

Table B.15: Simple censoring of the small weather model for spring. Rotated copulae are shown by the following degree. θ denotes the first copula parameter, and δ the second.

Tree	Copula	Family	θ	δ	p-value
T_1	$C_{I,W}$	Clayton 90°	0.42	–	0.82
	C_{I,T_D}	Frank	0.52	–	0.1
T_2	$C_{W,T_D\|I}$	Independence	–	–	–

Table B.16: Simple censoring of the small weather model for summer. Rotated copulae are shown by the following degree. θ denotes the first copula parameter, and δ the second.

Tree	Copula	Family	θ	δ	p-value
T_1	C_{W,T_M}	Frank	1.47	–	0.04
	C_{I,T_M}	Gumbel	1.15	–	0.98
T_2	$C_{W,I\|T_M}$	Clayton 270°	0.35	–	0.00

Table B.17: Simple censoring of the small weather model for fall. Rotated copulae are shown by the following degree. θ denotes the first copula parameter, and δ the second.

Appendix C

Code

In this section we present some of the R-code used for the analysis. The implementation follows a similar structure to `VineCopula` Schepsmeier et al. (2018). The code is written with the intention to be used for this analysis, not to be generally applicable to a wider audience. If the regular vine construction were to be implemented again, the spanning trees would likely be implemented from scratch, since the `igraph` library was a bit hard to use for this purpose. Note that the code provided is mainly the fully censored regular vine construction, which is an excerpt of the full implementation. Code for the simulation study, copula expression etc. can be found in a repository on GitHub under `JohanBirk/interval_censored_regular_vines`.

C.1 Fully Interval Censored Regular Vine Construction

```
library(tidyverse)
library(magrittr)
library(rlist)
library(igraph)
library(VineCopula)
library(copula)
library(doParallel)
library(foreach)
source(file = 'censor_est.R', local = TRUE)
source(file = 'copula_cdfs.R', local = TRUE)
source(file = "censor_gof_parallel.R")

log_lik_censor.intern <- function(u_upper, u_lower, v_upper, v_lower, cop, check.pars
    ){
  cop_fun <- cop$cop_fun
  cop_dens <- cop$cop_dens
  cop_du <- cop$cop_du
  cop_dv <- cop$cop_dv
  theta <- cop$theta
  delta <- cop$delta
  loglik <- 0

  # Observation not tied in either margin
  places1 <- ((u_lower == u_upper) & (v_lower == v_upper))
```

```
  loglik <- sum(log(
    cop_dens(u = u_upper[places1],v = v_upper[places1], theta, delta, check.pars =
        check.pars)
  ))

  # l smaller in both
  places2 <- ((u_lower < u_upper) & (v_lower < v_upper))
  loglik <- loglik + sum(log(
    cop_fun(u = u_upper[places2], v = v_upper[places2], theta, delta, check.pars =
        check.pars) -
      cop_fun(u = u_upper[places2], v = v_lower[places2], theta, delta, check.pars =
          check.pars) -
      cop_fun(u = u_lower[places2], v = v_upper[places2], theta, delta, check.pars =
          check.pars) +
      cop_fun(u = u_lower[places2], v = v_lower[places2], theta, delta, check.pars =
          check.pars)
  ))

  # u_lower smaller, v not tied
  places3 <- ((u_lower < u_upper) & (v_lower == v_upper))
  loglik <- loglik + sum(log(
    cop_dv(u = u_upper[places3], v = v_upper[places3], theta, delta, check.pars =
        check.pars) -
      cop_dv(u = u_lower[places3],v = v_upper[places3], theta, delta, check.pars =
          check.pars)
  ))

  # v_lower smaller, u not tied
  places4 <- ((u_lower == u_upper) & (v_lower < v_upper))
  loglik <- loglik + sum(log(
    cop_du(u = u_upper[places4], v = v_upper[places4], theta, delta, check.pars =
        check.pars) -
      cop_du(u = u_upper[places4], v = v_lower[places4], theta, delta, check.pars =
          check.pars)
  ))
  return(loglik)
}


censor_est.full <- function(u_upper, u_lower, v_upper, v_lower, cop_name, check.pars
    = FALSE){
  ## This function receives rotated data, and estimates the copula. Intended for use
      with fully censored R-vine.
  cop <- get_cop(cop_name)
  ## When conditional data are computed, the data computed from max rank estimated
      marginals are not always
  ## larger than when computed by min rank estimated marginals.
  u_up <- pmax(u_upper, u_lower)
  u_lo <- pmin(u_upper, u_lower)
  v_up <- pmax(v_upper, v_lower)
  v_lo <- pmin(v_upper, v_lower)

  if(cop$n.param == 1){
    ll <- function(theta){
      cop$theta <- theta
      return(log_lik_censor.intern(u_upper = u_up,u_lower = u_lo,
                                   v_upper = v_up, v_lower = v_lo, cop = cop, check.
                                       pars = check.pars))
    }
    optimlist <- list()
    objectives <- c()
    for(i in 1:length(cop$optim.limits)){
      cop$theta <- cop$optim.limits[[i]]$start
      optimlist %<>% rlist::list.append(optimize(f = ll, maximum = TRUE,
                                                 interval = c(cop$optim.limits[[i]]$
                                                     low,
                                                              cop$optim.limits[[i]]$
                                                                  up)))
```

```
77        objectives %<>% c(optimlist[[i]]$objective)
78      }
79      if(sum(is.na(objectives)) == length(objectives)){
80        warning("All NA produced by optimize. Narrowing the parameter search.")
81        ## Using inverse kendall to find better parameters
82
83        optimlist <- list()
84        objectives <- c()
85        lim_adjust <- 3
86        inv_tau <- VineCopula::BiCopTau2Par(family = cop$fam, cor(u_lower, v_lower,
              method = "kendall"))
87        if(cop$fam %in% c(3,4,6)){
88          inv_tau <- abs(inv_tau)
89        }
90        for(i in 1:length(cop$optim.limits)){
91          cop$theta <- cop$optim.limits[[i]]$start
92          optimlist %<>% rlist::list.append(optimize(f = ll, maximum = TRUE,
93                                                     interval = c(max(cop$optim.limits
                                                                         [[i]]$low, inv_tau - lim_
                                                                         adjust),
94                                                                  min(cop$optim.limits
                                                                         [[i]]$up, inv_
                                                                         tau + lim_adjust
                                                                         ))))
95          objectives %<>% c(optimlist[[i]]$objective)
96        }
97        if(sum(is.na(objectives)) == length(objectives)){
98          stop("All NA produced by optimize after narrowed search.")
99        }else{
100         cop$theta <- optimlist[[which.max(objectives)]]$maximum
101         cop$log.lik <- optimlist[[which.max(objectives)]]$objective
102         cop$AIC <- 2*cop$n.param - 2*optimlist[[which.max(objectives)]]$objective
103       }
104     }else{
105       cop$theta <- optimlist[[which.max(objectives)]]$maximum
106       cop$log.lik <- optimlist[[which.max(objectives)]]$objective
107       cop$AIC <- 2*cop$n.param - 2*optimlist[[which.max(objectives)]]$objective
108     }
109     return(cop)
110   }else{
111     cop$theta <- cop$start.value[1]
112     cop$delta <- cop$start.value[2]
113     parlower <- cop$optim.lower
114     parupper <- cop$optim.upper
115
116     ll <- function(par){
117       cop$theta <- par[1]
118       cop$delta <- par[2]
119       return(log_lik_censor.intern(u_upper = u_up,u_lower = u_lo,
120                                    v_upper = v_up, v_lower = v_lo,cop = cop, check.
                                         pars = check.pars))
121     }
122     optimout <- optim(par = c(cop$theta, cop$delta), fn = ll,
123                       method = "L-BFGS-B",
124                       lower = parlower, upper = parupper,
125                       control = list(fnscale = -1, maxit = 500))
126     cop$theta <- optimout$par[1]
127     cop$delta <- optimout$par[2]
128     cop$log.lik <- optimout$value
129     cop$AIC <- 2*cop$n.param - 2*optimout$value
130     return(cop)
131   }
132 }
133
134
135 censor_copula_select.full <- function(u_upper, u_lower, v_upper, v_lower, indeptest =
        FALSE, level = 0.05,
```

```
                                        include_tawn = TRUE, include_amh = TRUE,
                                            include_t = TRUE){
  tau <- cor(u_upper, v_upper, method = "kendall")
  N <- length(u_upper)
  if(indeptest){
    f <- sqrt((9 * N * (N - 1))/(2 * (2 * N + 5))) * abs(tau)
    p.value = 2 * (1 - pnorm(f))
    if(p.value>level){
      return(get_cop("Independence"))
    }
  }

  copula_estimates <- list()
  copula_aics <- c()
  ## Radially symmetric copulas
  copulas <- c("Frank", "normal", "t")
  if(!include_t){
    copulas <- copulas[-3]
  }
  for(cop_name in copulas){
    fitted_copula <- try(censor_est.full(u_upper, u_lower, v_upper, v_lower, cop_name
        ))
    if(class(fitted_copula) == "try-error"){
      print(cop_name)
      copula_estimates %<>% rlist::list.append(get_cop(cop_name))
      copula_aics %<>% c(Inf)
    }else{
      fitted_copula$rotation <- 1
      copula_estimates %<>% rlist::list.append(fitted_copula)
      copula_aics %<>% c(fitted_copula$AIC)
    }
  }
  ## 0 and 180 degree rotations
  if(tau>0){
    copulas <- c("AMH", "Clayton", "Gumbel", "Joe",
                 "BB1", "BB6", "BB7", "BB8", "Tawn", "Tawn2")
    if((tau>=1/3)| (!include_amh)){
      copulas <- copulas[-1]
    }
    if(!include_tawn){
      copulas <- copulas[!copulas %in% c("Tawn", "Tawn2")]
    }
  }else{
    copulas <- c("AMH", "Clayton")
    if((tau<=(5 - 8/log(2)) / 3)| (!include_amh)){
      copulas <- copulas[-1]
    }
  }

  for(cop_name in copulas){
    for(rot in c(1,3)){
      if(rot == 1){
        ## no rotation
        fitted_copula <- try(censor_est.full(u_upper = u_upper, u_lower = u_lower, v_
            upper = v_upper, v_lower = v_lower, cop_name))
      }else if(rot == 3){
        ## 180 degrees 1 - v, 1 - u
        fitted_copula <- try(censor_est.full(u_upper = 1 - u_lower, u_lower = 1 - u_
            upper, v_upper = 1 - v_lower, v_lower = 1 - v_upper, cop_name))
      }
      if(class(fitted_copula) == "try-error"){
        copula_estimates %<>% rlist::list.append(get_cop(cop_name))
        copula_aics %<>% c(Inf)
      }else{
        fitted_copula$rotation <- rot
        copula_estimates %<>% rlist::list.append(fitted_copula)
        copula_aics %<>% c(fitted_copula$AIC)
      }
```

```
    }
  }
  ## 90 and 270 degree rotations
  if(tau<0){
    copulas <- c("AMH", "Clayton", "Gumbel", "Joe",
                 "BB1", "BB6", "BB7", "BB8", "Tawn", "Tawn2")
    if((tau<=-1/3)| (!include_amh)){
      copulas <- copulas[-1]
    }
    if(!include_tawn){
      copulas <- copulas[!copulas %in% c("Tawn", "Tawn2")]
    }
  }else{
    copulas <- c("AMH", "Clayton")
    if((tau>=-(5 - 8/log(2)) / 3) | (!include_amh)){
      copulas <- copulas[-1]
    }
  }
  for(cop_name in copulas){
    for(rot in c(2,4)){
      if(rot == 2){
        ## 90 degrees 1 - u
        fitted_copula <- try(censor_est.full(u_upper = 1 - u_lower, u_lower = 1 - u_
            upper, v_upper = v_upper, v_lower = v_lower, cop_name))
      }else if(rot == 4){
        ## 270 degrees 1 - v
        fitted_copula <- try(censor_est.full(u_upper = u_upper, u_lower = u_lower, v_
            upper = 1 - v_lower, v_lower = 1 - v_upper, cop_name))
      }
      if(class(fitted_copula) == "try-error"){
        copula_estimates %<>% rlist::list.append(get_cop(cop_name))
        copula_aics %<>% c(Inf)
      }else{
        fitted_copula$rotation <- rot
        copula_estimates %<>% rlist::list.append(fitted_copula)
        copula_aics %<>% c(fitted_copula$AIC)
      }
    }
  }
  cop <- copula_estimates[[which.min(copula_aics)]]
  cop$tau <- tau
  return(cop)
}

rvine_translation_pval <- function(tree, names){
  tree_0 <- tree
  d <- length(tree) + 1
  R_matrix <- matrix(0, nrow = d, ncol = d)
  cop_matrix <- matrix(0, nrow = d, ncol = d)
  theta_matrix <- matrix(0, nrow = d, ncol = d)
  delta_matrix <- matrix(0, nrow = d, ncol = d)
  aic_matrix <- matrix(0, nrow = d, ncol = d)
  cvm_matrix <- matrix(0, nrow = d, ncol = d)
  ks_matrix <- matrix(0, nrow = d, ncol = d)

  # starting from top level of the tree
  for(i in (d-1):2){
    var <- tree[[i]]$t.cond[i,1]
    var %<>% c(tree[[i]]$t.cond[i,2])
    biCop <- cop_name2BiCop(tree[[i]]$Copulas[[1]])
    copulas <- c(biCop$Family)
    thetas <- c(biCop$Par)
    deltas <- c(biCop$Par2)
    aics <- c(tree[[i]]$Copulas[[1]]$AIC)
    cvms <- c(tree[[i]]$Copulas[[1]]$p.value.CvM)
    kss <- c(tree[[i]]$Copulas[[1]]$p.value.KS)
    for(j in (i-1):1){
      edge <- ceiling(which(tree[[j]]$t.cond[j,] == var[1])/2)
```

```
      ind <- which(tree[[j]]$t.cond[j,(edge*2 - 1):(2*edge)] != var[1]) - 1

      var %<>% c(tree[[j]]$t.cond[j,2*edge - 1 + ind])
      biCop <- cop_name2BiCop(tree[[j]]$Copulas[[edge]])
      copulas %<>% c(biCop$Family)
      thetas %<>% c(biCop$Par)
      deltas %<>% c(biCop$Par2)
      aics %<>% c(tree[[j]]$Copulas[[edge]]$AIC)
      cvms %<>% c(tree[[j]]$Copulas[[edge]]$p.value.CvM)
      kss %<>% c(tree[[j]]$Copulas[[edge]]$p.value.KS)

      # Removing used entries
      tree[[j]]$t.cond <- matrix(tree[[j]]$t.cond[,-c(edge*2 - 1, edge*2)], nrow=j)
      tree[[j]]$Copulas[[edge]] <- NULL
    }
    R_matrix[(d+1-length(var)):d,d - i] <- var
    cop_matrix[(d + 2 -length(var)):d,d - i] <- copulas
    theta_matrix[(d + 2 -length(var)):d,d - i] <- thetas
    delta_matrix[(d + 2 -length(var)):d,d - i] <- deltas
    aic_matrix[(d + 2 -length(var)):d,d - i] <- aics
    cvm_matrix[(d + 2 -length(var)):d,d - i] <- cvms
    ks_matrix[(d + 2 -length(var)):d,d - i] <- kss
  }
  ind <- !(tree[[2]]$t.cond[2,] %in% diag(R_matrix))
  var <- rev(tree[[2]]$t.cond[,ind])
  R_matrix[(d+1-length(var)):d,d - 1] <- var
  R_matrix[d,d] <- var[2]

  biCop <- cop_name2BiCop(tree[[1]]$Copulas[[1]])
  cop_matrix[d,d-1] <- biCop$Family
  theta_matrix[d,d-1] <- biCop$Par
  delta_matrix[d,d-1] <- biCop$Par2
  aic_matrix[d,d-1] <- tree[[1]]$Copulas[[1]]$AIC
  cvm_matrix[d,d-1] <- tree[[1]]$Copulas[[1]]$p.value.CvM
  ks_matrix[d,d-1] <- tree[[1]]$Copulas[[1]]$p.value.KS

  return(list(RVine = RVineMatrix(Matrix = R_matrix,
                                  family = cop_matrix,
                                  par = theta_matrix,
                                  par2 = delta_matrix,
                                  names = names),
              AICs = aic_matrix,
              p.value.CvM = cvm_matrix,
              p.value.KS = ks_matrix,
              Tree = tree_0))
}

## These are modified since the upper limit becomes the lower under rotation.
transform_u.full <- function(u_upper, u_lower, v_upper, v_lower, cop, method){
  rot <- cop$rotation
  if(method == "upper"){
    # rot == 1 is 0 degrees, and no rotation
    if(rot == 2){
      # 90 degree
      return(cop$cop_du(u = 1 - u_lower, v = v_upper, theta = cop$theta, delta = cop$
          delta))
    }else if(rot == 3){
      # 180 degree
      return(1 - cop$cop_du(u = 1 - u_lower, v = 1 - v_lower, theta = cop$theta,
          delta = cop$delta))
    }else if(rot == 4){
      # 270 degree
      return(1 - cop$cop_du(u = u_upper, v = 1 - v_lower, theta = cop$theta, delta =
          cop$delta))
    }
    return(cop$cop_du(u = u_upper, v = v_upper, theta = cop$theta, delta = cop$delta)
        )
  }else{
```

```
    # rot == 1 is 0 degrees, and no rotation
    if(rot == 2){
      # 90 degree
      return(cop$cop_du(u = 1 - u_upper, v = v_lower, theta = cop$theta, delta = cop$
          delta))
    }else if(rot == 3){
      # 180 degree
      return(1 - cop$cop_du(u = 1 - u_upper, v = 1 - v_upper, theta = cop$theta,
          delta = cop$delta))
    }else if(rot == 4){
      # 270 degree
      return(1 - cop$cop_du(u = u_lower, v = 1 - v_upper, theta = cop$theta, delta =
          cop$delta))
    }
    return(cop$cop_du(u = u_lower, v = v_lower, theta = cop$theta, delta = cop$delta)
        )
  }
}

transform_v.full <- function(u_upper, u_lower, v_upper, v_lower, cop, method){
  rot <- cop$rotation
  if(method == "upper"){
    # rot == 1 is 0 degrees, and no rotation
    if(rot == 2){
      # 90 degree
      return(cop$cop_dv(u = 1 - u_lower, v = v_upper, theta = cop$theta, delta = cop$
          delta))
    }else if(rot == 3){
      # 180 degree
      return(1 - cop$cop_dv(u = 1 - u_lower, v = 1 - v_lower, theta = cop$theta,
          delta = cop$delta))
    }else if(rot == 4){
      # 270 degree
      return(1 - cop$cop_dv(u = u_upper, v = 1 - v_lower, theta = cop$theta, delta =
          cop$delta))
    }
    return(cop$cop_dv(u = u_upper, v = v_upper, theta = cop$theta, delta = cop$delta)
        )
  }else{
    # rot == 1 is 0 degrees, and no rotation
    if(rot == 2){
      # 90 degree
      return(cop$cop_dv(u = 1 - u_upper, v = v_lower, theta = cop$theta, delta = cop$
          delta))
    }else if(rot == 3){
      # 180 degree
      return(1 - cop$cop_dv(u = 1 - u_upper, v = 1 - v_upper, theta = cop$theta,
          delta = cop$delta))
    }else if(rot == 4){
      # 270 degree
      return(1 - cop$cop_dv(u = u_lower, v = 1 - v_upper, theta = cop$theta, delta =
          cop$delta))
    }
    return(cop$cop_dv(u = u_lower, v = v_lower, theta = cop$theta, delta = cop$delta)
        )
  }
}

censor_RVine_select_full <- function(data, indeptest = TRUE, level = 0.05,
                                     include_tawn = TRUE, include_amh =
                                         FALSE, include_t = TRUE, core_lim
                                         =20,
                                     calc_pVal = FALSE, N_bootstrap = NA){
  data_upper <- pobs(data, ties.method = "max")
  data_lower <- pobs(data, ties.method = "min")
  d <- ncol(data)
  n <- nrow(data)
  if(is.na(N_bootstrap)){
```

```
    N_bootstrap <- 10*n
  }
  tree <- list()
  copulas <- list()

  ## Constructing a matrix of sources, destinations and weights of each edge
  ## that can be converted to an igraph like graph
  tau <- cor(data_upper, method = "kendall")
  weights <- tau[upper.tri(tau)]
  sources <- matrix(rep(1:d, d), ncol = d)
  sources <- sources[upper.tri(sources)]
  destinations <- matrix(rep(1:d, d), ncol = d, byrow = TRUE)
  destinations <- destinations[upper.tri(destinations)]
  totals <- cbind(abs(weights), sources, destinations) ## Total of weights, sources
      and destinations
  totals <- totals[order(totals[,1], decreasing = TRUE),]
  graph <- graph_from_edgelist(cbind(as.character(totals[,2]), as.character(totals
      [,3])), directed = FALSE)
  E(graph)$weight <- -as.numeric(totals[,1])
  graph <- minimum.spanning.tree(graph)

  ################### transforming ########################
  transformed_upper <- matrix(0, nrow=n, ncol = 2*length(E(graph)))
  transformed_lower <- matrix(0, nrow=n, ncol = 2*length(E(graph)))
  transformed_cond <- matrix(0, nrow=1, ncol = 2*(d-1))
  for(j in 1:length(E(graph))){
    ind1 <- as.numeric(tail_of(graph, j)$name)
    ind2 <- as.numeric(head_of(graph, j)$name)
    ui_upper <- data_upper[,ind1]
    ui_lower <- data_lower[,ind1]
    vi_upper <- data_upper[,ind2]
    vi_lower <- data_lower[,ind2]
    cop <- censor_copula_select.full(ui_upper, ui_lower, vi_upper, vi_lower,
                                     indeptest = indeptest, level = level,
                                     include_tawn = include_tawn, include_amh =
                                         include_amh, include_t = include_t)
    if(calc_pVal){
      test <- censor_gof_test_parallel(ui_upper, vi_upper, cop = cop, N = N_bootstrap
          , core_lim = core_lim)
      cop$p.value.CvM <- test$p.value.CvM
      cop$p.value.KS <- test$p.value.KS
    }

    transformed_upper[,(j*2-1)] <- transform_u.full(u_upper = ui_upper, u_lower =  ui
        _lower, v_upper = vi_upper, v_lower = vi_upper,
                                                    cop = cop, method = "upper")
    transformed_lower[,(j*2-1)] <- transform_u.full(u_upper = ui_upper, u_lower =  ui
        _lower, v_upper = vi_upper, v_lower = vi_upper,
                                                    cop = cop, method = "lower")
    transformed_upper[,(j*2)] <- transform_v.full(u_upper = ui_upper, u_lower =  ui_
        lower, v_upper = vi_upper, v_lower = vi_upper,
                                                  cop = cop, method = "upper")
    transformed_lower[,(j*2)] <- transform_v.full(u_upper = ui_upper, u_lower =  ui_
        lower, v_upper = vi_upper, v_lower = vi_upper,
                                                  cop = cop, method = "lower")
    transformed_cond[1,(j*2-1)] <- ind1
    transformed_cond[1,(j*2)] <- ind2
    copulas %<>% rlist::list.append(Copula = cop)
  }
  tree %<>% rlist::list.append(list(Copulas = copulas, Graph = graph,
                                    t.data_upper = transformed_upper, t.data_lower =
                                        transformed_lower, t.cond = transformed_cond
                                    ))

  ############## Transformation on higher levels
  for(i in 2:(d-1)){
    # Building structure for the next full tree
    tau <- cor(transformed_upper, method = "kendall")
```

```
442     sources <- c()
443     destinations <- c()
444     weights <- c()
445     vertex_data_upper <- matrix(nrow=n)
446     vertex_data_lower <- matrix(nrow=n)
447     vertex_cond <- matrix(nrow=i-1)
448     vertex_var <- c()
449     ## Looping over all edges in the previous tree to find all possible options for
            the next tree given this
450     for(j in 1:(length(E(graph))-1)){
451       for(k in (j+1):length(E(graph))){
452         first_ends <- ends(graph, j, names = FALSE)
453         sec_ends <- ends(graph, k, names = FALSE)
454         if(any(first_ends %in% sec_ends)){
455           sources %<>% c(paste(ends(graph, j, names = TRUE), collapse = ","))
456           destinations %<>% c(paste(ends(graph, k, names = TRUE), collapse = ","))
457
458           if(i>=3){
459             ## This if/else is simply due to compatibility issues with the graph from
                    the first tree
460             ## For T_3 and so on, I have better control over the conditioned
                    variables and the conditioning sets
461             ## and which edges go where. This is a more tedious process for finding
                    this.
462             first_involved <- unique(c(transformed_cond[1:i-1,(j*2-1):(j*2)]))
463             second_involved <- unique(c(transformed_cond[1:i-1,(k*2-1):(k*2)]))
464             conditioned <- unique(first_involved[first_involved %in% second_involved
                    ])
465             cond1 <- which((transformed_cond[i-1,(j*2-1):(j*2)] %in% conditioned)) +j
                    *2-2
466             cond2 <- which((transformed_cond[i-1,(k*2-1):(k*2)] %in% conditioned)) +k
                    *2-2
467             weights %<>% c(tau[cond1, cond2])
468
469             vertex_data_upper %<>% cbind(transformed_upper[,cond1], transformed_upper
                    [,cond2])
470             vertex_data_lower %<>% cbind(transformed_lower[,cond1], transformed_lower
                    [,cond2])
471             vertex_cond %<>% cbind(conditioned, conditioned)
472             ind1 <- which(!(transformed_cond[i-1,(j*2-1):(j*2)] %in% conditioned)) +j
                    *2-2
473             ind2 <- which(!(transformed_cond[i-1,(k*2-1):(k*2)] %in% conditioned)) +k
                    *2-2
474             vertex_var %<>% c(transformed_cond[i-1,ind1] , transformed_cond[i-1,ind2
                    ])
475           }else{
476             ind1 <- which(first_ends %in% sec_ends) + j*2 - 2
477             ind2 <- which(sec_ends %in% first_ends) + k*2 - 2
478             cond1 <- which(!(first_ends %in% sec_ends)) + j*2 - 2
479             cond2 <- which(!(sec_ends %in% first_ends)) + k*2 - 2
480
481             weights %<>% c(tau[ind1, ind2])
482             vertex_data_upper %<>% cbind(transformed_upper[,ind1], transformed_upper
                    [,ind2])
483             vertex_data_lower %<>% cbind(transformed_lower[,ind1], transformed_lower
                    [,ind2])
484
485
486             vertex_cond %<>% cbind(transformed_cond[,ind1], transformed_cond[,ind2])
487             vertex_var %<>% c(transformed_cond[i-1,cond1], transformed_cond[i-1,cond2
                    ])
488           }
489         }
490       }
491     }
492     vertex_data_upper <- vertex_data_upper[,-1]
493     vertex_data_lower <- vertex_data_lower[,-1]
494     vertex_cond <- matrix(vertex_cond[,-1], nrow=i-1)
```

```
    graph <- graph_from_edgelist(cbind(sources, destinations), directed = FALSE)
    E(graph)$weight <- -abs(weights)
    new_graph <- minimum.spanning.tree(graph)

    ########################################################
    ## In this step, we find the index to which edges have been removed to construct the spanning tree
    ## I did not find a "natural" way of doing this with the igraph library, but this tedious solution works
    if(length(E(graph)) != length(E(new_graph))){
      graph_names <- vector("character", length = length(E(graph)))
      for(j in 1:length(E(graph))){
        graph_names[j] <- paste(ends(graph,j), collapse = " ")
      }
      new_graph_names <- vector("character", length = length(E(new_graph)))
      for(j in 1:length(E(new_graph))){
        new_graph_names[j] <- paste(ends(new_graph,j), collapse = " ")
      }
      data_removal <- which(!(graph_names %in% new_graph_names))
      vertex_data_upper <- vertex_data_upper[,-c(data_removal*2 - 1, data_removal*2)]
      vertex_data_lower <- vertex_data_lower[,-c(data_removal*2 - 1, data_removal*2)]
      vertex_cond <- vertex_cond[,-c(data_removal*2 - 1, data_removal*2)]
      vertex_var <- vertex_var[-c(data_removal*2 - 1, data_removal*2)]
    }
    ########################################################
    ########################################################

    graph <- new_graph
    copulas <- list()
    transformed_upper <- matrix(0, nrow = n, ncol = 2*length(E(graph)))
    transformed_lower <- matrix(0, nrow = n, ncol = 2*length(E(graph)))
    for(j in 1:length(E(graph))){
      ind1 <- j*2 - 1
      ind2 <- j*2
      ui_upper <- vertex_data_upper[,ind1]
      ui_lower <- vertex_data_lower[,ind1]
      vi_upper <- vertex_data_upper[,ind2]
      vi_lower <- vertex_data_lower[,ind2]
      cop <- censor_copula_select.full(ui_upper, ui_lower, vi_upper, vi_lower,
                                       indeptest = indeptest, level = level,
                                       include_tawn = include_tawn,
                                       include_amh = include_amh,
                                       include_t = include_t)
      if(calc_pVal){
        test <- censor_gof_test_parallel(ui_upper, vi_upper, cop = cop, N = N_bootstrap, core_lim = core_lim)
        cop$p.value.CvM <- test$p.value.CvM
        cop$p.value.KS <- test$p.value.KS
      }
      transformed_upper[,ind1] <- transform_u.full(u_upper = ui_upper, u_lower = ui_lower, v_upper = vi_upper, v_lower = vi_lower, cop = cop, method = "upper")
      transformed_lower[,ind1] <- transform_u.full(u_upper = ui_upper, u_lower = ui_lower, v_upper = vi_upper, v_lower = vi_lower, cop = cop, method = "lower")
      transformed_upper[,ind2] <- transform_v.full(u_upper = ui_upper, u_lower = ui_lower, v_upper = vi_upper, v_lower = vi_lower, cop = cop, method = "upper")
      transformed_lower[,ind2] <- transform_v.full(u_upper = ui_upper, u_lower = ui_lower, v_upper = vi_upper, v_lower = vi_lower, cop = cop, method = "lower")
      copulas %<>% rlist::list.append(Copula = cop)
    }
    transformed_cond <- rbind(vertex_cond, vertex_var)
    tree %<>% rlist::list.append(list(Copulas = copulas, Graph = graph,
                                      t.data_upper = transformed_upper, t.data_lower = transformed_lower, t.cond = transformed_cond))
```

```
550     }
551     return(rvine_translation_pval(tree = tree, names = colnames(data)))
552 }
```

Johan Østebø Birketvedt

Interval Censored Regular Vines With Application to Event-Based Modelling of Precipitation and Temperature

NTNU
Kunnskap for en bedre verden