# Study of Copula and Its Applications

*A Project report submitted in partial fulfilment of the requirements for the Degree of M.Sc. (Statistics) with specialization in Industrial Statistics*

*Submitted by*

**Miss. Badgujar Monika Sanjay** (366334)

**Miss. Patil Kamini Ananda** (366351)

**Mr. Patil Ratnesh Dnyaneshwar** (366356)

*Under the guidance of*

**Asst. Prof. MANOJ. C. PATIL**

**DEPARTMENT OF STATISTICS**
**SCHOOL OF MATHEMATICAL SCIENCES**
**KAVAYITRI BAHINABAI CHAUDHARI**
**NORTH MAHARASHTRA UNIVERSITY**
**JALGAON – 425001**

**(2024–2025)**

# DEPARTMENT OF STATISTICS
# KAVAYITRI BAHINABAI CHAUDHARI
# NORTH MAHARASHTRA UNIVERSITY, JALGAON

## CERTIFICATE

This is to certify that **Miss. Badgujar Monika Sanjay**, **Miss. Patil Kamini Ananda**, **Mr. Patil Ratnesh Dnyaneshwar** students of M.Sc.(Statistics) with specialization in Industrial Statistics, at Kavayitri Bahinabai Chaudhari, North Maharashtra University, Jalgaon have successfully completed their project work entitled “Study of Copula and Its Applications” as a part of M.Sc. (Statistics) program under my guidance and supervision during the academic year 2024-2025.

**Asst. Prof. Manoj C. Patil**

(Project Guide)

Signature: ............

# Abstract

This project report, titled **"Study of Copula and Its Applications,"** investigates the mathematical framework of copulas, which are vital for modeling dependencies between random variables. Utilizing Sklar's Theorem, the study illustrates how copulas can decompose multivariate distributions into their marginal distributions and a copula function that captures the underlying dependence structure. Various types of copulas, including Gaussian, Clayton, and Gumbel, are categorized and discussed in terms of their properties and suitability for different applications.

The report highlights two significant applications of copulas: in hydrology for analyzing rainfall data and in multivariate process control. In the context of hydrology, copulas are employed to model the dependencies between different rainfall variables, allowing for a comprehensive understanding of rainfall distribution and variability, which is crucial for water resource management and flood risk assessment. Additionally, the report explores the development of copula-based control charts in multivariate process control, enhancing traditional monitoring techniques by accommodating non-linear dependencies and effectively detecting small shifts in process means.

This dual focus on theoretical foundations and practical applications underscores the versatility of copulas in addressing real-world challenges in statistical modeling and decision-making.

# Acknowledgement

We would like to express our sincere gratitude to Prof. K. K. Kamalja, Head of the Department of Statistics, Kavayitri Bahinabai Chaudhari, North Maharashtra University, Jalgaon, for her unwavering support and invaluable comments. Her generous attitude and guidance have been instrumental in making our work much easier and more meaningful.

We extend our heartfelt appreciation to our project guide, Mr. M. C. Patil, for his help, inspiration, and guidance throughout the project. Under his supervision, we have gained a wealth of knowledge and skills, and his support during challenging moments made the project journey smoother.

We are sincerely thankful for his continuous support. We would also like to acknowledge the valuable guidance provided by Prof. R. L. Shinde and Dr. R. D. Koshti, during the project work. Their expertise and insights have significantly enriched our understanding and contributed to the success of our project.

We are grateful to our friends and the entire non-teaching staff of the Department of Statistics for their assistance and for providing us with the necessary laboratory and other facilities.

Place: Jalgaon
Date:

Miss. Badgujar Monika Sanjay (Seat no: 366334)
Miss. Patil Kamini Ananda (Seat no: 366351)
Mr. Patil Ratnesh Dnyaneshwar (Seat no: 366356)

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction to Copulas

## 1.1 Introduction

A copula is a function that links univariate marginal distributions to form a multivariate distribution. According to Nelsen [2006] in *An Introduction to Copulas*, Sklar's Theorem plays a pivotal role in the theory of copulas by asserting that any multivariate distribution can be decomposed into its marginal distributions and a copula, which captures the dependence structure between the variables. This decomposition allows researchers to model the marginal behavior of each variable separately from their joint dependence, providing greater flexibility in statistical modeling.

Copulas are particularly useful in fields where understanding the joint behavior of variables is crucial, such as finance, insurance, and risk management. They enable the modeling of complex dependencies, including tail dependence, which refers to the occurrence of extreme values in multiple variables simultaneously—a critical aspect in risk analysis. Several types of copulas, including Gaussian, Clayton, and Gumbel, offer different ways to model dependencies, each with specific properties suited to various applications.

The use of copulas has become widespread in recent years, not only because of their theoretical elegance but also due to their practical applicability in real-world problems where traditional correlation measures may not suffice. By employing copulas, analysts can achieve a more nuanced understanding of the dependencies between variables, leading to better decision-making in fields such as finance, engineering, and environmental science.

## 1.2 Copula Function

### What are Copulas?

The term *Copula* comes from the Latin word *copulare* which means "a link, tie, bond", Nelsen [2006] referring to joining together. With this meaning, a copula is defined as a function that joins multivariate distribution functions to their one-dimensional marginal distribution functions. It is a multivariate distribution function defined on the unit $n$-cube $[0,1]^n$, with uniformly distributed marginals for more study to refer Kpanzou [2007].

**Definition 2.1.** An $n$-dimensional copula is a function $C : [0,1]^n \to [0,1]$, with the following properties:

1. $C$ is grounded, which means that for every $\mathbf{u} = (u_1, u_2, \ldots, u_n) \in [0,1]^n$, $C(\mathbf{u}) = 0$ if at least one coordinate $u_i$ is zero, $i = 1, 2, \ldots, n$,

2. $C$ is $n$-increasing, which means that for every $\mathbf{u} \in [0,1]^n$ and $\mathbf{v} \in [0,1]^n$ such that $\mathbf{u} \leq \mathbf{v}$, the $C$-volume $V_C([\mathbf{u},\mathbf{v}])$ of the box $[\mathbf{u},\mathbf{v}]$ is non-negative,

3. $C(1,\ldots,1,u_i,1,\ldots,1) = u_i$, for all $u_i \in [0,1]$, $i = 1,2,\ldots,n$.

**Definition 2.2.** For a bivariate distribution, the copula is a function $C_{XY}$ defined by:

$$C_{XY}(F_X(x),F_Y(y)) = P(X \leq x, Y \leq y) \tag{1.1}$$

This is often written in the more compact form:

$$C(u,v) = F_{XY}(xy) \tag{1.2}$$

where $u = F_X(x)$ and $v = F_Y(y)$. This definition can be extended to the multivariate case where we have:

$$C(u_1,u_2,\ldots,u_d) = F_{X_1,X_2,\ldots,X_d}(X_1,X_2,\ldots,X_d) \tag{1.3}$$

where $u_i = F_{X_i}(x_i)$

## 1.3 Families of Copula

In statistical and probabilistic modeling, a copula is a function that links multivariate distribution functions to their one-dimensional marginal distribution functions. Various families of copulas are used to model different types of dependency structures between variables. Here's a brief overview of some common families of copulas discussed in their paper Kpanzou [2007]

### 1.3.1 The Farlie-Gumbel-Morgenstern (FGM) Family

$$C_\theta(u,v) = uv + \theta uv(1-u)(1-v) \tag{1.4}$$

, where $\theta \in [-1,1]$ These are the only coulas whose functional form is a polynomial quadratic in u and in v.they are commonaly denoted FGM copulas.
Members of the FGM family are symmetric, i.e $C_\theta(u,v) = C_\theta(u,v)$ for all (u,v) in $I^2$. A pair (X,Y) of random variable is said to be exchangable if the vectors (X,Y) and (Y,X) are identically distributed.

### 1.3.2 Cuadras-Augé family of copulas

Let $\theta \in [0,1]$. The function $C_\theta$ defined by

$$C_\theta(u,v) = \frac{[\min(u,v)]^\theta}{[uv]^{1-\theta}} = \begin{cases} uv^{1-\theta}, & \text{if } u \leq v, \\ u^{1-\theta}v, & \text{if } u \geq v, \end{cases} \tag{1.5}$$

is a copula function. This family is known as the Cuadras-Augé family of copulas.

### 1.3.3 Marshal-Olkin family

If $\alpha,\beta \in [0,1]$, then the function $C_{\alpha,\beta} : [0,1]^2 \to [0,1]$, defined by

$$C_{\alpha,\beta}(u,v) = \min(u^{1-\alpha}v, uv^{1-\beta}), \tag{1.6}$$

is a bivariate copula function. This two-parameter family of copulas is the Marshal-Olkin family.

### 1.3.4 Normal Family

Let $N_\rho(x,y)$ denote the standard bivariate normal joint distribution function with correlation coefficient $\rho$. Then, the copula $C_p$ corresponding to $N_\rho$ is given by:

$$C_p(u,v) = N_\rho\left(\Phi^{-1}(u), \Phi^{-1}(v)\right), \tag{1.7}$$

where $\Phi$ is the CDF of the standard normal distribution, and $\Phi^{-1}$ is its inverse function.

## 1.4 Some Properties of Copulas

**Sklar's Theorem Nelsen [2006]:**
The importance of copulas in statistics is described in Sklar's theorem. In this sense, this theorem is considered the central theorem of copula theory.
**Sklar's Theorem in n-dimensions:**

Let $F$ be a joint (cumulative) distribution function with marginal cumulative distribution functions $F_1, F_2, \ldots, F_n$. Then there exists a copula $C$ such that for all $x_1, x_2, \ldots, x_n \in [-\infty, \infty]$:

$$F_X(x_1, x_2, \ldots, x_n) = C(F_{X_1}(x_1), F_{X_2}(x_2), \ldots, F_{X_n}(x_n)) \tag{1.8}$$

In the case of variables that have a continuous distribution, the copula is unique.
**Sklar's Theorem in two-dimensions:**

Let $F$ be a joint distribution function with marginals $F(x)$ and $F(y)$. There exists a copula $C$ such that for all $x$ and $y$ in $\mathbb{R}$:

$$F(x,y) = C(F(x), F(y)) \tag{1.9}$$

### 1.4.1 Properties of Copulas

1. **A copula is an increasing function of its inputs:**

   $$C(u_1, \ldots, u_i^*, \ldots, u_d) > C(u_1, \ldots, u_i, \ldots, u_d)$$

   for $u_i^* > u_i$ and $i = 1, \ldots, d$. This makes sense from a probabilistic perspective because, if $u_i^* > u_i$, then

   $$P(X_i \leq x_i*) > P(X_i \leq x_i) \text{ for corresponding } x_i* = F_{X_i}^{-1}(u_i*) \text{ and } x_i = F_{X_i}^{-1}(u_i),$$

   and hence:

   $$P(X_1 \leq x_1, \ldots, X_i \leq x_i*, \ldots, X_d \leq x_d) > P(X_1 \leq x_1, \ldots, X_i \leq x_i*, \ldots, X_d \leq x_d).$$

2. **If all the marginal CDFs are equal to 1 except for one of them, then the copula function is equal to the value of that one marginal CDF:**

   $$C(1, \ldots, u_i, 1, \ldots, 1) = U_i for i = 1, \ldots, d and U_i \in [0,1]$$

   This makes sense because $u_K = 1 \implies P(X \leq x_k) = 1$ (i.e a certainty), for $X_k = F^{-1}(u_K)$, and the only uncertainty in the above joint probability is the marginal probability with respect to the i-th variable.

3. **A copula function always returns a valid probability:**

   $$C(u_1, u_2, \ldots, u_d) \in [0,1]$$

### 1.4.2 Frechet-Hoeffding Bounds

By Dep [2017], For every copula $C$ and every $(u, v) \in [0,1]^2$,

$$\max(u + v - 1, 0) \leq C(u, v) \leq \min(u, v). \tag{1.10}$$

$W(u, v) = \max(u + v - 1, 0)$ and $M(u, v) = \min(u, v)$ are themselves copulas.

## 1.5 Copulas and Association

This section contain different ways in which copulas can be used in the study of dependence between random variables introduce in Shemyakin and Kniazev [2017]

1. **Kandall's Tau:**
   Kendall's tau measure of a pair (X, Y ), distributed according to joint distribution function, is defined as the difference between the probabilities of concordance and discordance for two independent pairs (X1, Y1) and (X2, Y2) each with distribution function; that is

$$\tau \ = P\left[(X_1 - X_2)(Y_1 - Y_2) > 0\right] - P\left[(X_1 - X_2)(Y_1 - Y_2) < 0\right] \tag{1.11}$$

2. **Spearman's Rho:**
   Let (X1, Y1), (X2, Y2) and (X3, Y3) be three independent random vectors, copies of a random vector (X, Y), with a common joint distribution function . The Spearman's rho associated with (X, Y ), distributed according to distribution function, is defined by

$$\rho \ = 3P\left[(X_1 - X_2)(Y_1 - Y_3) > 0\right] - P\left[(X_1 - X_2)(Y_1 - Y_3) < 0\right] \tag{1.12}$$

   Spearman's rho can be expressed in terms of the copula C associated with the joint distribution.

$$\rho \ = 12 \int_0^1 \int_0^1 \left[C(u, v) - uv\right] du\, dv \tag{1.13}$$

3. **Tail Dependence:**

   Tail dependence refers to the relationship between the extreme values (tails) of two random variables.

- **Upper Tail Dependence Coefficient ($\lambda_U$):** The upper tail dependence coefficient measures the degree of dependence between the upper tails of the distributions of two randomvariables X and Y. It is defined as:

$$\lambda_U = \lim_{u \to 1} P(X \geq F^{-1}(u) \mid Y \geq F^{-1}(u)) \tag{1.14}$$

  Coefficient of upper tail dependence in terms of the copula function:

$$\lambda_U = \lim_{u \to 1} \frac{C(u, v)}{1 - u} \tag{1.15}$$

- **Lower Tail Dependence Coefficient ($\lambda_L$):**
  The lower tail dependence coefficient measures the degree of dependence between the lower tails of the distributions of X and Y. It is defined as:

$$\lambda_L = \lim_{u\to 0^+} P\left(X \le F_X^{-1}(u) \mid Y \le F_Y^{-1}(u)\right) \tag{1.16}$$

  Coefficient of lower tail dependence in terms of the copula function:

$$\lambda_L = \lim_{u\to 0^+} \frac{C(u,u)}{u} \tag{1.17}$$

### Survival Copula

The survival copula is a function that describes the joint survival probabilities of two random variables, X and Y. It is defined in terms of the survival functions of these variables. The survival function $F_X(x)$ is given by:
$F_X(x) = P(X > x) = 1 - F_X(x)$
$simillary, F_Y(y) = P(Y > y) = 1 - F_Y(y)$
The survival copula $C(u,v)$ is defined as:

$$C(u,v) = P(X > x, Y > y) = C(F_X(x), F_Y(y))$$

where $u = F_X(x)$ and $v = F_Y(y)$.
The relationship between the original copula $C$ and the survival copula $\widehat{C}$ is given by:

$$\widehat{C}(u,v) = 1 - C(1-u, 1-v) \tag{1.18}$$

The coefficient of upper tail dependence $\lambda_u$ in terms of the survival copula function is defined as:

$$\lambda_u = \lim_{u\to 1^+} \frac{P(X > F_X^{-1}(u) \mid Y > F_Y^{-1}(u))}{u} \tag{1.19}$$

Alternatively, using the survival copula $\widehat{C}$, it can be expressed as:

$$\lambda_u = \lim_{u\to 1^+} \frac{1 - 2u + \widehat{C}(u,u)}{1-u} \tag{1.20}$$

## 1.6 Types of Copula

By Shemyakin and Kniazev [2017], There are three main families of copula :

1. fundamental copulas

2. explicit copulas

3. implicit copulas

### 1.6.1 Fundamental Copulas

A fundamental copula is a specific type of copula that represents the three basic forms of dependency that can exist between random variables, namely:

- Independence
- perfect positive interdependence
- perfect negative interdependence.

There are three type of fundamental copulas:

1. Independence (or product) copula
2. Co-monotonic (or minimum) copula
3. Counter-monotonic (or maximum) copula

**Independence (or product) copula**

This copula represents the case where two random variables are independent of each other. The joint distribution of the variables is simply the product of their individual distributions. Mathematically, it is expressed as:

$$C(u,v) = uv \tag{1.21}$$

Here we have:

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

$$P(X \leq x, Y \leq y) = C[F_X(x), F_Y(y)]$$

This captures the property of independence of the two variables X and Y, and so is also called the independence (or product) copula.

**Co-monotonic (or minimum) copula**

This copula is used where random variable demonstrate perfect positive interdependence. The co-monotonic copula is defined in the bivariate case as:

$$C[u,v] = \min(u,v) \tag{1.22}$$

Here we have:

$$C[F_X(x), F_Y(y)] = \min(F_X(x), F_Y(y))$$

or

$$P(X \leq x, Y \leq y) = \min(P(X \leq x), P(Y \leq y))$$

#### Counter-monotonic (or maximum) copula

The counter-monotonic copula captures the relationship between two variables whose values are perfectly positively interdependent on each other, while the counter-monotonic copula capturesthe correspondinginverse relationship.
The counter-monotonic copula is defined in the bivariate case as:

$$C[u,v] = \max(u+v-1,0) \tag{1.23}$$

Here, we have:

$$C[F_X(x),F_Y(y)] = \max(F_X(x)+F_Y(y)-1,0)$$

or

$$P(X \le x, Y \le y) = \max(P(X \le x)+P(Y \le y)-1,0)$$

### 1.6.2 Explicit Copula

An explicit copula is a type of copula that has a simple closed-form of mathematical expression, allowing for straightforward representation and calculation of the joint distribution of random variables based on their marginal distributions. Below are three examples of commonly used explicit copulas

#### Gumbel copula

The Gumbel copula is a specific type of copula used to model the dependence structure between two random variables, particularly focusing on uppertail dependence.
The Gumbel copula is defined in the bivariate case as

$$C(u,v) = \exp\left(-\left[(-\ln u)^{\alpha} + (-\ln v)^{\alpha}\right]^{1/\alpha}\right) \tag{1.24}$$

where $\alpha$ is a parameter that influences the strength of dependence.
Note that the Gumbel copula is often referred to as the Gumbel - Hougaard copula.

#### Clayton copula

The Clayton copula is another important type of copula used to model the dependence structure between two random variables, particularly focusing on lower tail dependence The Clayton copula is defined in the bivariate case as:

$$C(u,v) = \left(u^{-\alpha} + v^{-\alpha} - 1\right)^{-1/\alpha} \tag{1.25}$$

where $\alpha$ is a parameter that influences the strength of dependence.

#### Frank copula

The Frank copula describes an interdependence structure in which there is no upper or lower tail dependence. The Frank copula is defined in the bivariate case as:

$$C(u,v) = -\frac{1}{\alpha}\ln\left[1 + \frac{(e^{-\alpha u}-1)(e^{-\alpha v}-1)}{e^{-\alpha}-1}\right] \tag{1.26}$$

**Archimedean copulas:**

Archimedean copulas are a class of copulas that can be defined using a generator function. In the bivariate case, they take the form:

$$C(u,v) = \psi^{-1}(\psi(u) + \psi(v)) \tag{1.27}$$

where $\psi(x)$ is the generator function, and $\psi^{-1}$ is the pseudo-inverse function.
Archimedean copulas are a subset of explicit copulas. The Gumbel, Clayton, and Frank copulas are all examples of Archimedean copulas.
The definition of Archimedean copulas can be extended to more than 2 dimensions. The copula is expressed as:

$$C(u_1, u_2, \ldots, u_d) = \psi^{-1}\left(\sum_{i=1}^{d} \psi(u_i)\right) \tag{1.28}$$

The generator function $\psi : (0,1] \to [0,\infty]$ must be continuous, strictly decreasing, and convex, with $\psi(1) = 0$.
Some important families of Archimedean copulas are given in Table 1.2.

**Pseudo-inverse functions**

In order to define Archimedean copulas, the concept of a pseudo-inverse function is introduced to extend the idea of an inverse function. The pseudo-inverse function $\psi^{-1}(x)$ of a function $\psi(x)$ is defined as follows
Where:
$\psi^{-1}(x)$ denotes the ordinary inverse function obtained by inverting the equation $\psi(x) = y$ to express $y$ in terms of $x$.
The function $\psi(x)$ is a continuous, strictly decreasing, convex function that maps the interval $(0,1)$ to a finite range.
If $\psi(0) = \infty$, the pseudo-inverse is always equal to the 'ordinary' inverse, and the generator function is called a strict generator function.

### 1.6.3 Implicit Copula

Implicit copulas are based on multivariate distributions, such as the multivariate normal distribution or the multivariate Student's $t$ distribution. **The Gaussian copula and the Student's $t$-copula** are examples of implicit copulas.

**Gaussian copula**

The Gaussian copula is one of the most commonly used implicit copulas and is based on the multivariate normal distribution.It is defined in the bivariate case as follows:

$$C(u,v) = \Phi_p\left(\Phi^{-1}(u), \Phi^{-1}(v)\right) \tag{1.29}$$

**Student's $t$-copula**

The Student's t copula is another important type of implicit copula, which is based on the multivariate Student'st distribution.It is defined in the bivariate case as follows:

$$C(u,v) = t_v\left(t^{-1}(u), t^{-1}(v)\right) \tag{1.30}$$

Table 1.1: Families of Bivariate Extreme Value Copulas

| **Model** | $\boldsymbol{A_\theta(t)}$ | $\boldsymbol{C_{A_\theta}(u,v)}$ |
|---|---|---|
| Gumbel | $\theta t^2 - \theta t + 1,$ | $uv\exp(-\theta\log(u)\log(v)\log(uv))$ |
| | $\theta \in (0,1)$ | |
| Gumbel-Hougaard | $[t^{\frac{1}{1-\theta}} + (1-t)^{\frac{1}{1-\theta}}]^{1-\theta},$ | $\exp\left(-\left[\lvert\log(u)\rvert^{\frac{1}{1-\theta}} + \lvert\log(v)\rvert^{\frac{1}{1-\theta}}\right]^{1-\theta}\right)$ |
| | $\theta \in (0,1)$ | |
| Galambos | $1-[t^{-\theta} + (1-t)^{-\theta}]^{-1/\theta},$ | $uv\exp\left(-\left[\lvert\log(u)\rvert^{-\theta} + \lvert\log(v)\rvert^{-\theta}\right]^{-1/\theta}\right)$ |
| | $\theta \in (0,\infty)$ | |
| Generalised Marshall-Olkin | $\max\{1-\theta_1 t, 1-\theta_2(1-t)\},$ | $u^{1-\theta_1} v^{1-\theta_2}\min(u^{\theta_1}, v^{\theta_2})$ |
| | $(\theta_1,\theta_2) \in (0,1)^2$ | |

Table 1.2: Families of Bivariate Archimedean Copulas

| **Family** | **Generator** $\varphi(t)$ | **Bivariate Copula** $C_\varphi(u,v)$ |
|---|---|---|
| Independence | $-\log(t)$ | $uv$ |
| Clayton<br>(Cook-Johnson, Oakes) | $\varphi(t) = \frac{t^{-\alpha}-1}{\alpha},$<br>$\alpha \in (0,\infty)$ | $(u^{-\alpha} + v^{-\alpha} - 1)^{-\frac{1}{\alpha}}$ |
| Gumbel<br>(Hougaard) | $\varphi(t) = (-\log(t))^\alpha,$<br>$\alpha \in [1,\infty)$ | $\exp\left(-\left[(-\log(u))^\alpha + (-\log(v))^\alpha\right]^{\frac{1}{\alpha}}\right)$ |
| Frank | $\varphi(t) = \log\left(\frac{e^{\alpha t}-1}{e^\alpha - 1}\right),$<br>$\alpha \in \mathbb{R} \setminus \{0\}$ | $\frac{1}{\alpha}\log(1 + (e^{\alpha u} - 1)(e^{\alpha v} - 1)e^{-\alpha})$ |

# Chapter 2

# Estimation and Simulation of copula

## 2.1 Estimation

An estimation approach is proposed for models for a multivariate response with covariates when each of the parameters (either univariate or a dependence parameter) of the model can be associated with a marginal distribution. In this there are many methods to estimate a copula. Introduce in Shemyakin and Kniazev [2017]

### 2.1.1 Inference Method for Margins

By Kpanzou [2007], Inference Functions for Margins (IFM) is a two-step estimation method commonly used in copula modeling. It separates the estimation of the marginal distributions from the estimation of the copula.

Consider a random vector $\mathbf{X} = (X_1, X_2, \ldots, X_d)$ consisting of $d$ continuous random variables. The joint distribution of $\mathbf{X}$ can be expressed using Sklar's Theorem as: For more detail refer Nelsen [2006]

$$\mathbf{F}_{\mathbf{X}}(x_1, x_2, \ldots, x_d) = C\left(F_1(x_1), F_2(x_2), \ldots, F_d(x_d); \theta\right) \tag{2.1}$$

### Step 1: Estimation of Marginal Distributions

- The first step involves estimating the parameters of the marginal distributions independently. Assume each marginal distribution $F_i(x_i; \alpha_i)$ has a parameter vector $\alpha_i$.

- For each marginal distribution $F_i(x_i)$, you calculate the log-likelihood function based on the observed data:

  $$L_i(\alpha_i) = \sum_{j=1}^{n} \log f_i(x_{ij}; \alpha_i) \tag{2.2}$$

  where $f_i(x_{ij}; \alpha_i)$ is the probability density function corresponding to the distribution $F_i(x_i)$, and $x_{ij}$ are the observed values of $X_i$.

- Maximize the log-likelihood function $L_i(\alpha_i)$ with respect to $\alpha_i$ to obtain the maximum likelihood estimates (MLEs) $\hat{\alpha}_i$ for each marginal distribution.

  $$\hat{\alpha}_i = \arg\max_{\alpha_i} \sum_{j=1}^{n} \log f_i(x_{ij}; \alpha_i) \tag{2.3}$$

### Step 2: Estimating Copula Parameters

- Once the marginal parameters $\hat{\alpha}_1, \hat{\alpha}_2, \dots, \hat{\alpha}_d$ are estimated, you move on to estimate the parameters of the copula, $\theta$.

- The log-likelihood function for the joint distribution is then:

$$L(\hat{\alpha}_1, \hat{\alpha}_2, \dots, \hat{\alpha}_d; \theta) = \sum_{j=1}^{n} \log \left[ c\left(F_1(x_{1j}; \hat{\alpha}_1), \dots, F_d(x_{dj}; \hat{\alpha}_d); \theta\right) \prod_{i=1}^{d} f_i(x_{ij}; \hat{\alpha}_i) \right] \tag{2.4}$$

  where $c(\cdot)$ is the copula density function, and $\prod_{i=1}^{d} f_i(x_{ij}; \hat{\alpha}_i)$ accounts for the marginal densities.

- Maximize this function with respect to $\theta$ to obtain the estimate $\hat{\theta}$.

$$\hat{\theta} = \arg\max_{\theta} \sum_{j=1}^{n} \log \left[ c\left(F_1(x_{1j}; \hat{\alpha}_1), \dots, F_d(x_{dj}; \hat{\alpha}_d); \theta\right) \prod_{i=1}^{d} f_i(x_{ij}; \hat{\alpha}_i) \right] \tag{2.5}$$

### Example:

### Estimating Parameters for a Bivariate Copula using Inference Method for margin.

Suppose a bivariate random vector $\mathbf{X} = (X_1, X_2)$ with marginal distributions $F_1(x_1)$ and $F_2(x_2)$. Then

$$X_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$$

$$X_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$$

We assume the joint distribution of $\mathbf{X}$ is modeled by a Gaussian copula with an unknown correlation parameter $\rho$.Then we estimate parameter using IFM metod.
First, we estimate the parameters of the marginal distributions independently.
For the first marginal distribution $F_1(x_1)$: The log-likelihood function is:

$$L_1(\mu_1, \sigma_1) = \sum_{j=1}^{n} \log \left[ \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left( -\frac{(x_{1j} - \mu_1)^2}{2\sigma_1^2} \right) \right] \tag{2.6}$$

Simplifying, we maximize:

$$L_1(\mu_1, \sigma_1) = -\frac{n}{2} \log(2\pi) - n \log(\sigma_1) - \frac{1}{2\sigma_1^2} \sum_{j=1}^{n} (x_{1j} - \mu_1)^2 \tag{2.7}$$

The maximum likelihood estimates (MLEs) for $\mu_1$ and $\sigma_1$ are:

$$\hat{\mu}_1 = \frac{1}{n} \sum_{j=1}^{n} x_{1j}$$

$$\hat{\sigma}_1^2 = \frac{1}{n} \sum_{j=1}^{n} (x_{1j} - \hat{\mu}_1)^2$$

Similarly, for the second marginal distribution $F_2(x_2)$, the MLEs are:

$$\hat{\mu}_2 = \frac{1}{n}\sum_{j=1}^{n} x_{2j}$$

$$\hat{\sigma}_2^2 = \frac{1}{n}\sum_{j=1}^{n} (x_{2j} - \hat{\mu}_2)^2$$

Next, using the estimated marginal parameters $\hat{\mu}_1, \hat{\sigma}_1, \hat{\mu}_2, \hat{\sigma}_2$, we estimate the copula parameter $\rho$. For a Gaussian copula, the joint log-likelihood function is:

$$L(\rho) = \sum_{j=1}^{n} \log\left[\frac{1}{\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)}\left(Z_{1j}^2 + Z_{2j}^2 - 2\rho Z_{1j} Z_{2j}\right)\right)\right] \tag{2.8}$$

where

$$Z_{1j} = \frac{x_{1j} - \hat{\mu}_1}{\hat{\sigma}_1}$$

and

$$Z_{2j} = \frac{x_{2j} - \hat{\mu}_2}{\hat{\sigma}_2}$$

Maximizing this log-likelihood with respect to $\rho$, we obtain the MLE $\hat{\rho}$.
To find the MLE $\hat{\rho}$, we take the derivative of $L(\rho)$ with respect to $\rho$ and set it to zero:

$$\frac{\partial L(\rho)}{\partial \rho} = 0$$

The derivative of the log-likelihood function with respect to $\rho$ is:

$$\frac{\partial L(\rho)}{\partial \rho} = \frac{n\rho}{1-\rho^2} - \frac{1}{1-\rho^2}\sum_{j=1}^{n} Z_{1j} Z_{2j} + \frac{\rho(1-\rho^2)}{2}\sum_{j=1}^{n}\left(Z_{1j}^2 + Z_{2j}^2 - 2\rho Z_{1j} Z_{2j}\right) \tag{2.9}$$

then

$$\frac{\partial L(\rho)}{\partial \rho} = 0$$

yields:

$$\frac{n\rho}{1-\rho^2} = \frac{1}{1-\rho^2}\sum_{j=1}^{n} Z_{1j} Z_{2j} \tag{2.10}$$

Simplifying further, we get:

$$n\rho = \sum_{j=1}^{n} Z_{1j} Z_{2j}$$

Therefore, the MLE for $\rho$ is:

$$\hat{\rho} = \frac{1}{n}\sum_{j=1}^{n} Z_{1j} Z_{2j} \tag{2.11}$$

### 2.1.2 The Maximum Likelihood Method

By Choroś et al., The Maximum Likelihood Estimation for copula parameters involves deriving the joint density from the copula and marginal distributions, forming the log-likelihood function, and then numerically maximizing this function to obtain the parameter estimates.

1. **Specify the Copula Model** Choose the appropriate copula model
2. **Collect and Transform Data** Collect a sample
$$\{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\},$$
where $n$ is the number of observations. Transform the original data into pseudo-observations
$$\{(U_1, V_1), (U_2, V_2), \ldots, (U_n, V_n)\}$$
using the empirical cumulative distribution functions (CDFs):
$$U_i = \hat{F}_n(X_i),$$
$$V_i = \hat{G}_n(Y_i),$$
for $i = 1, 2, \ldots, n$, where $\hat{F}_n$ and $\hat{G}_n$ are the empirical CDFs of $X$ and $Y$, respectively.
3. **Construct the Log-Likelihood Function** Joint density of the copula
$$c(u, v; \theta) = \frac{\partial^2 C(u, v; \theta)}{\partial u \partial v}$$
The log-likelihood function for the parameter $\theta$
$$L(\theta) = \sum_{i=1}^{n} \log c(U_i, V_i; \theta)$$
4. **Maximize the Log-Likelihood Function** to find the parameter $\theta$ that maximizes the log-likelihood function.
$$\hat{\theta} = \arg\max_{\theta} L(\theta)$$
since the log-likelihood function does not have a closed-form solution.Then we use a numerical optimization method such as Newton-Raphson to estimate $\hat{\theta}$

### 2.1.3 The Empirical Copula Function

The Empirical Copula Function is a non-parametric method used to estimate the copula function, which describes the dependence structure between random variables, independently of their marginal distributions. This method is particularly useful because it does not assume any specific parametric form for the copula, instead relying on the data itself to estimate the copula function. In their paperDurante and Sempi [2010]

### Empirical Distribution Function (EDF):

Consider a sample $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$ which are independent and identically distributed (iid) of a random vector $(X, Y)$. The bivariate empirical distribution function $H_n(x, y)$ is defined as:

$$H_n(x, y) = \frac{1}{n} \sum_{i=1}^{n} 1\{X_i \leq x, Y_i \leq y\} \tag{2.12}$$

Here, $1\{\cdot\}$ is the indicator function, which is 1 if the condition inside is true and 0 otherwise.

## Empirical Copula Function:

The empirical copula function $C_n(u,v)$ is defined using the empirical distribution functions. It captures the joint dependence structure of the variables $X$ and $Y$ after transforming their marginal distributions to uniform distributions on the interval $[0,1]$.

$$C_n(u,v) = H_n\left(F_n^{-1}(u), G_n^{-1}(v)\right) \tag{2.13}$$

This expression involves the inverse empirical distribution functions $F_n^{-1}(u)$ and $G_n^{-1}(v)$, which map uniform random variables $u$ and $v$ back to the original scale of $X$ and $Y$. Another way to express this is:

$$C_n(u,v) = \frac{1}{n}\sum_{k=1}^{n} 1\{X_k \leq F_n^{-1}(u), Y_k \leq G_n^{-1}(v)\} \tag{2.14}$$

## Estimating the Parameter of a Copula Using the Empirical Copula Function.

1. **Generate or Obtain Data**

   Consider a sample of size $n$ from a bivariate distribution $(X_i, Y_i)$, where $i = 1, 2, \ldots, n$.

2. **Compute Empirical Marginal Distribution Functions**
   Calculate the empirical distribution functions for the marginals X and Y. These functions estimate the marginal distributions from the sample data.

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} I\{X_i \leq x\}$$

$$G_n(y) = \frac{1}{n}\sum_{i=1}^{n} I\{Y_i \leq y\}$$

3. **Transform Data to Uniform Margins**
   Transform the sample data to uniform [0,1] random variables using the empirical distribution functions:

$$U_i = F_n(X_i), \quad V_i = G_n(Y_i)$$

4. **Construct the Empirical Copula Function**
   The empirical copula function $C_n(u,v)$ captures the joint dependence structure between $U$ and $V$. It is defined as:

$$C_n(u,v) = \frac{1}{n}\sum_{k=1}^{n} I\{U_k \leq u, V_k \leq v\}$$

5. **Specify the Theoretical Copula Model:**
   Assume a copula model $C(u,v;\theta)$ with a parameter $\theta$. For the Gumbel copula, the theoretical copula function is:

$$C(u,v;\theta) = \exp\left(-\left[\left((-\log u)^{\theta} + (-\log v)^{\theta}\right)^{1/\theta}\right]\right)$$

6. **Estimate the Copula Parameter**
   To estimate the parameter $\theta$, we minimize the discrepancy between the empirical copula function $C_n(u,v)$ and the theoretical copula function $C(u,v;\theta)$. To estimate the parameter $\theta$ in a copula model, we often use a least-squares approach.

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{n} \sum_{j=1}^{n} \left( C_n\left(\frac{i}{n}, \frac{j}{n}\right) - C\left(\frac{i}{n}, \frac{j}{n}; \theta\right) \right)^2$$

### 2.1.4 Method of Moments

The Method of Moments (MoM) is a classical technique for parameter estimation in statistical models. In the context of copulas, it involves estimating the copula parameters by equating theoretical expectations (derived from the copula model) with sample moments (or measures) calculated from the data. The Method of Moments for copulas generally involves the following steps:

1. **Select a Dependence Measure:** Choose a dependence measure that can be expressed as a function of the copula parameter(s). Common choices are:
   - **Kendall's tau** ($\tau$): Measures the concordance between pairs of observations.
   - **Spearman's rho** ($\rho$): Measures the rank correlation between variables.
2. **Derive Theoretical Expressions:** Determine the theoretical relationship between the chosen dependence measure and the copula parameter(s). This relationship is derived from the definition of the copula and the dependence measure.

   For example, in the case of the **Clayton copula** with parameter $\theta$, Kendall's tau $\tau$ is related to $\theta$ by:

$$\tau = \frac{\theta}{\theta + 2} \tag{2.15}$$

3. **Compute the Sample Dependence Measure:** Calculate the empirical value of the dependence measure (e.g., sample Kendall's tau) using the data. This step involves computing the measure from observed data pairs $(X_i, Y_i)$.
4. **Solve for the Parameter(s):** Equate the empirical dependence measure with the theoretical expression and solve for the copula parameter(s). This yields the method of moments estimator for the parameter.

   Continuing with the Clayton copula example:

$$\hat{\theta} = \frac{2\hat{\tau}}{1 - \hat{\tau}} \tag{2.16}$$

   where $\hat{\tau}$ is the sample estimate of Kendall's tau.

## Example:

## Estimating parameter for Clayton Copula using method of moment

1. **Define the Clayton Copula:** The Clayton copula is a popular copula for modeling asymmetric dependence, especially in cases where there is greater dependence in the lower tails.

$$C(u,v;\theta) = \left(\max\left(u^{-\theta} + v^{-\theta} - 1, 0\right)\right)^{-1/\theta} \tag{2.17}$$

where $\theta > 0$ controls the strength of dependence. As $\theta$ increases, the dependence between $X$ and $Y$ becomes stronger.

2. **Relationship Between Kendall's Tau and $\theta$:** For the Clayton copula, the theoretical relationship between Kendall's tau $\tau$ and the parameter $\theta$ is given by:

$$\tau = \frac{\theta}{\theta + 2} \tag{2.18}$$

3. **Estimation Using Sample Data:** Assume you have a bivariate dataset with $n$ pairs of observations $(X_i, Y_i)$.

   - Compute the sample Kendall's tau $\hat{\tau}$ from the data.
   - For example, suppose $\hat{\tau} = 0.4$.

4. **Solve for $\theta$:**
   Equate the sample Kendall's tau with the theoretical expression and solve for $\theta$:

$$\hat{\theta} = \frac{2\hat{\tau}}{1-\hat{\tau}} = \frac{2 \times 0.4}{1-0.4} = \frac{0.8}{0.6} \approx 1.33 \tag{2.19}$$

Thus, the estimate for $\theta$ is 1.33.

Table 2.1: Parameter Estimation Methods for Various Copulas using Simulation 5.1.

| Copula | IFM | MLE | Method of Moments |
|---|---|---|---|
| Gaussian | -0.04484735 | -0.01748052 | 0.008885545 |
| Clayton | 0.02974726 | 0.05963593 | 0.01137749 |
| Gumbel | 1 | 1.01 | 1.005689 |
| Frank | -0.03419829 | 0.02408141 | 0.05091334 |
| Student-t | -0.04949582 | -0.01650621 | 0.008885196 |

### 2.1.5 Adaptive Maximization by Parts

For this research, the adaptive maximization by parts (AMBP) proposed by Kim et al. [2007]. was applied to estimate the copula dependence estimator. Estimate the initial parameters $(\alpha_1, \beta_1, \theta_1)$ using the IFM method:

$$\alpha_1, \beta_1 = \arg\max\left(l_m(\alpha, \beta)\right)$$

$$\theta_1 = \arg\max\left(l_c(\alpha_1, \beta_1, \theta)\right) \tag{2.20}$$

$$\alpha_k, \beta_k = \arg\max\left(l(\alpha, \beta, \theta_{k-1})\right)$$

$$\theta_k = \arg\max\left(l_c(\alpha_k, \beta_k, \theta)\right) \tag{2.21}$$

for $k = 2, 3, 4, \ldots$ The IFM estimators $(\hat{\alpha}, \hat{\beta}, \hat{\theta})$ are taken as the initial values of the parameters $(\alpha_1, \beta_1, \theta_1)$ in Step 1 for the AMBP steps. While, for Step $k$, the $\theta$ in $l(\alpha, \beta, \theta)$ is replaced with $\theta_{k-1}$ and then the log-likelihood equation is maximized with respect to the marginal parameters $(\alpha, \beta)$ to estimate the next $(\alpha_k, \beta_k)$.

After that, same as Step 2 in the IFM method, $\alpha$ and $\beta$ in the copula log-likelihood model, $l_c(\alpha,\beta,\theta)$ are replaced with estimators of $\alpha_k$ and $\beta_k$ to estimate the next $\theta_k$. As the number $k$ tends to infinity, the estimator converges to the MLE of $(\alpha,\beta,\theta)$. The estimation performance of these three parametric methods is compared through the simulation and empirical studies.

## 2.2 Simulation

Simulation is a widely used tool, using the power of the computer, for experimenting with complex stochastic models. Simulation of copulas refers to the process of generating random samples from a multivariate distribution to study the dependence structure between random variables.

### Purpose of Simulation

Simulating copulas is often used to understand and analyze the joint behavior of random variables, especially in fields like finance, insurance, and risk management. By simulating data from a copula, researchers can explore how different dependence structures affect the behavior of the variables.

### 2.2.1 Inverse Transform Sampling

This method involves generating uniform random variables and then applying the inverse of the copula function to obtain samples from the desired joint distribution.As result Gliga [2014]

### Steps for Inverse Transform Sampling Method for Copulas

Copula Simulation Procedure

1. **Define the Copula and its Parameter:**
   Identify the copula function $C(u, v; \theta)$ where $\theta$ is the parameter of the copula. For instance, for a Gumbel copula, the copula function is:

$$C(u, v) = \exp\left(-\left[(-\ln u)^{\alpha} + (-\ln v)^{\alpha}\right]^{1/\alpha}\right)$$

2. **Generate Uniform Random Variables:**
   Generate $n$ samples of uniform random variables $U_i$ and $V_i$ from the interval $[0, 1]$. These will be used to generate the copula samples.

3. **Compute the Pseudo-Observations:**
   Compute the pseudo-observations using the empirical CDFs. For each $i$-th observation:

$$U_i = F_n(X_i) \qquad V_i = G_n(Y_i)$$

   where $F_n$ and $G_n$ are the empirical CDFs of the marginals $X$ and $Y$, respectively.

4. **Simulate from the Copula Function:**
   Use the inverse transform sampling method to generate samples from the copula function. For a given copula, compute the inverse of the copula function to find the required joint distribution.

5. **Transform to Original Marginals:**
   Finally, transform the generated uniform variables back to the original scale using the inverse CDFs of the marginal distributions. For instance:

$$X_i = F_X^{-1}(U_i) \qquad Y_i = F_Y^{-1}(V_i)$$

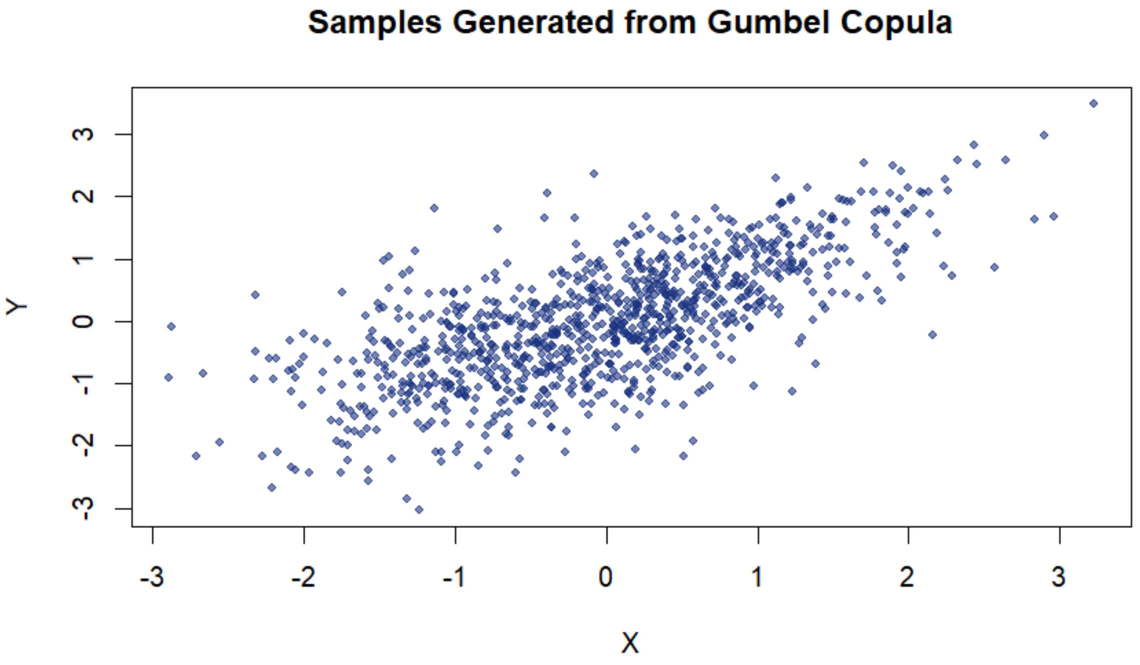

Figure 2.1: Sample Generated from Gumbel copula

### 2.2.2 Monte Carlo Simulation

Monte Carlo simulation is a statistical technique used to understand the behavior of a system or process by generating random samples from a defined probability distribution. In the context of copulas, Monte Carlo simulation is used to generate random samples from a copula, which models the dependence structure between two or more random variables. This allows for the analysis and understanding of how these variables interact and depend on each other which is well discussed in their paperShemyakin and Kniazev [2017].

## Simulating Data from Clayton Copula using Monte carlo simulation.

Clayton Copula Simulation Steps

1. **Define the Clayton Copula:**
   The Clayton copula function with parameter $\theta$ is:

$$C(u,v;\theta) = \left(u^{-\theta} + v^{-\theta} - 1\right)^{-\frac{1}{\theta}}$$

2. **Generate Uniform Random Variables:**
   Generate $n$ samples $U_i$ and $V_i$ from a uniform distribution on $[0,1]$:

3. **Simulate from the Copula:**
   Use the Clayton copula function to create dependent uniform random variables. To generate $U_i$ and $V_i$ according to a copula's dependency structure:

4. **Transform to Original Marginals:**
   Suppose $X$ and $Y$ have marginal distributions with CDFs $F_X$ and $F_Y$. Transform the uniform random variables to the original scale as follows:

$$X_i = F_X^{-1}(U_i) \qquad Y_i = F_Y^{-1}(V_i)$$

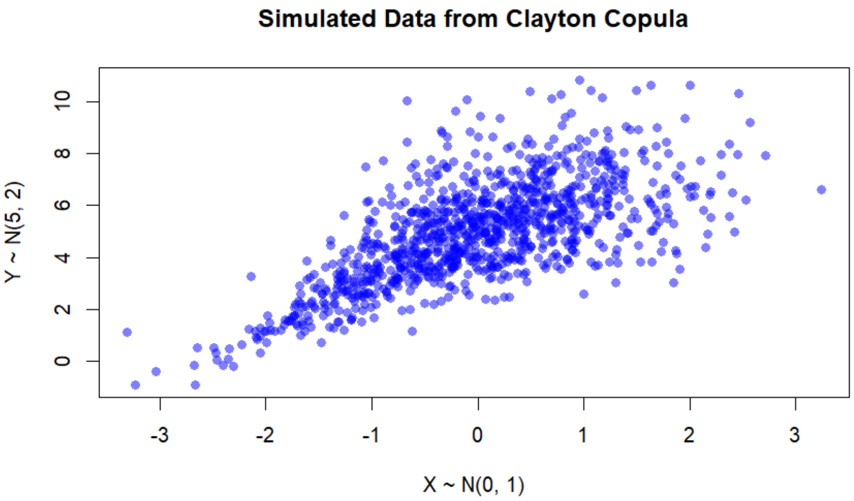

Figure 2.2: Simulated Data from Clayton copula

### 2.2.3 Conditional Simulation

Conditional simulation in copula models involves generating data based on specific conditions or constraints imposed on the copula's joint distribution. This technique is useful for simulating scenarios in which the joint behavior of variables needs to satisfy particular conditions, such as fixed values or ranges for some variables. Conditional simulation in copula models involves:

Steps for Copula Analysis

1. **Fitting the Copula:** Estimate the copula parameters using observed data.
2. **Defining Conditions:** Specify which variables are fixed and their values.
3. **Generating Samples:** Simulate data from the copula model while conditioning on the fixed variables.
4. **Analyzing Results:** Evaluate the behavior of the remaining variables under the given conditions.

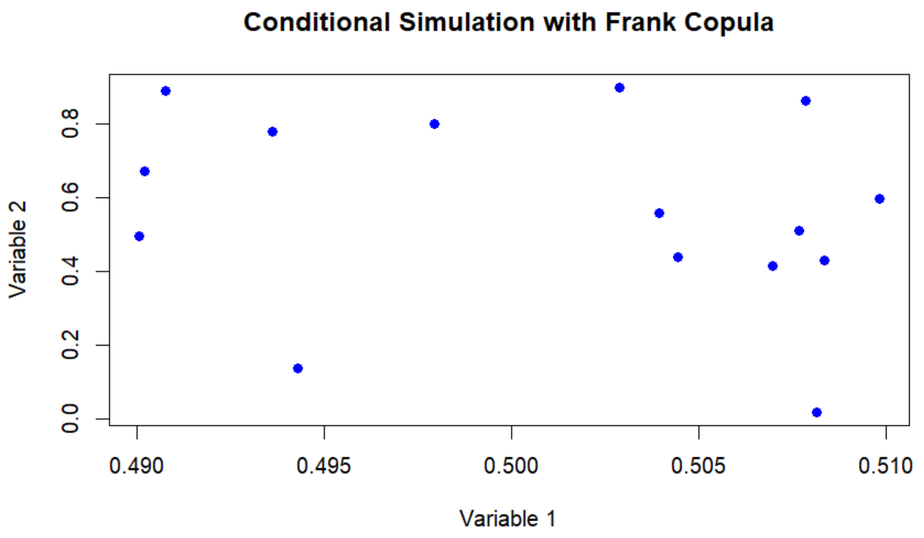

Figure 2.3: Conditional Simulation with Frank copula

### 2.2.4 Bootstrap simulation

Bootstrap simulation is a resampling technique used to estimate the distribution of a statistic or model parameter by repeatedly sampling from the original data set. In the context of copula models, this method helps assess the variability and reliability of the copula parameters and other related statistics. Here's a step-by-step explanation of how bootstrap simulation is applied in copula models:

Bootstrap Procedure

1. **Fit the Original Model:** Estimate the copula parameters using your actual dataset.
2. **Generate Bootstrap Samples:** Resample from the original dataset with replacement to create multiple bootstrap datasets.
3. **Refit the Model:** For each bootstrap sample, refit the copula model and estimate the parameters.

4. **Analyze and Summarize:** Analyze the distribution of bootstrap estimates to understand parameter variability and obtain confidence intervals.

## Using Bootstrap Simulation to Simulate sample from Gaussian Copula

To simulate samples from a Gaussian copula, Then following algorithm will generate samples that exhibit a specified correlation structure using a Gaussian copula and then transform those samples to uniform marginals.

Algorithm Steps

1. **Define the Correlation Matrix:** If the correlation coefficient $\rho$ is given, construct the correlation matrix $\Sigma$.
2. **Compute the Cholesky Decomposition:** To compute the Cholesky decomposition of the correlation matrix $\Sigma$ to obtain the lower triangular matrix $L$: $\Sigma = LL^T$.
3. **Generate Independent Standard Normal Samples:** Generate $n_{\text{samples}}$ of independent standard normal random variables $Z$: $Z \sim \mathcal{N}(0, I)$
4. **Introduce Correlation:** Transform the independent normal samples $Z$ to correlated samples $X$ using the Cholesky factor $L$:$X = ZL^T$
5. **Convert to Uniform Marginals:** Apply the CDF of the normal distribution to the correlated samples $X$ to obtain uniform marginals $U$: $U = \Phi(X)$

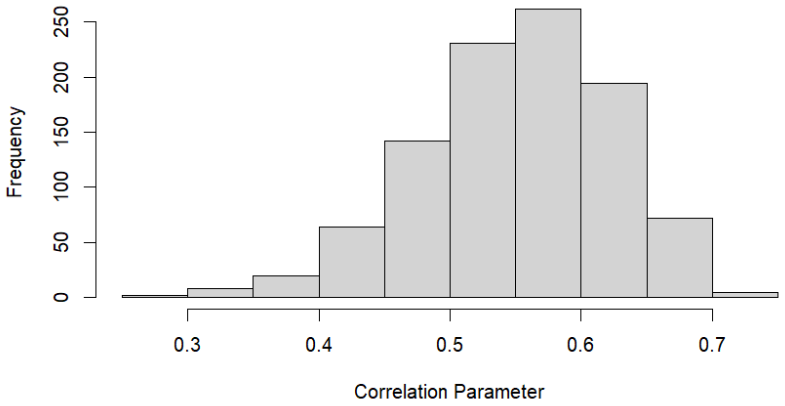

Figure 2.4: Bootsrap Distribution of Gaussian copula

# Chapter 3

# Application of Copulas in Rainfall Analysis

## Abstract

This study examines parametric methods: maximum likelihood (ML), inference function of margins (IFM), and adaptive maximization by parts (AMBP) for estimating copula dependence parameters through simulation and empirical analyses. Using 30 years of annual rainfall data from Dharangaon and Amalner stations , results indicate that IFM performs best for small sample sizes with correlation levels below 0.80, whereas AMBP excels for larger sample sizes across all correlation levels. These findings underscore the importance of estimating hydrological dependence structures, enabling the Pune Meteorological Department to generate hydrological events for improved flood and drought management systems.

## 3.1 Introduction

The copula method, introduced by SklarNelsen [2006], allows for the construction of joint distribution functions from uniform marginal distributions, overcoming traditional limitations by enabling the specification of any marginal distribution and copula for dependence structures. Zhang and Singh [2007] demonstrated its effectiveness in deriving bivariate joint distributions of rainfall variables with different marginals, without requiring normality or independence. Various copula families, including Archimedean, Gaussian, and Student's t, have been developed to cover diverse dependence structures. Parameter estimation methods for copulas are categorized into parametric, semiparametric, and nonparametric approaches, with studies comparing their performances.

In parametric approaches, marginal distributions are assumed to follow a specific parametric distribution, focusing on estimating marginal and copula dependence parameters. Three primary estimation methods are used: maximum likelihood (ML) estimation, inference function of margins (IFM), and maximization by parts (MBP). ML estimation is a direct method that simultaneously estimates marginal and copula parameters, making it the most efficient for the copula dependence parameter due to its asymptotic normality and consistency. However, maximizing the log-likelihood function can be challenging in practice, especially with high-dimensional parameters, leading to computational difficulties.

inference function of margins (IFM) method, which estimates marginal and copula parameters separately, maintaining efficiency similar to ML while enhancing computational practicality. The IFM method is particularly useful when ML estimation becomes too complex, focusing on implementation rather than theoretical analysis.The main advantage of this IFM method is it is computationally efficient than ML estimation because it does not estimate the marginal and dependence parameters simultaneously.IFM is efficient when bivariate random variables exhibit low dependecy.

In estimating copula parameters, particularly addressing the limitations of the Inference Function of Margins (IFM) method, which can lose efficiency by estimating marginal parameters without considering variable correlations. To enhance this process, Song et al. [2005] introduced the Maximization by Parts (MBP) method, which decomposes the full log-likelihood function into a working model (focusing on marginal parameters) and an error model (including both marginal and copula parameters). This approach simplifies computations but can be time-consuming and challenging with highly correlated variables or large sample sizes. To further improve efficiency. proposed the Adaptive Maximization by Parts (AMBP) algorithm, based on Meta t distributions. Additionally, the Kendall's tau method is noted as a popular semiparametric approach in hydrological analysis due to its straightforward relationship between rank correlation and copula parameters, making it easier to estimate. Overall, parametric approaches that consider marginal parameters are emphasized as yielding more precise estimates than semiparametric methods, which often overlook these parameters, thus enhancing the accuracy of dependency modeling in hydrological data.In hydrological analysis, the most common parametric approaches for estimating copula parameters are Maximum Likelihood (ML) estimation and Inference Function of Margins (IFM). However, studies that implement adaptive maximization by parts (AMBP) are a typical to find in hydrologic application in their paper Lokoman and Yusof [2018].

Therefore, this study focuses on the application of parametric approaches: maximum likelihood (ML) estimation, inference function of margins (IFM) and adaptive maximization by parts (AMBP) in estimating the copula dependence parameter. The estimation performance of the three parametric estimation methods is compared in the simulation and empirical studies

## 3.2 Literature Review

Fatahi et al. [2011] mean tested six copula models on rainfall data from Peninsular Malaysia (1980–2011). They discovered that these models are better than traditional methods for capturing how rainfall variables depend on each other, making them useful tools for understanding complex rainfall patterns. Similarly, Lokoman and Yusof [2018] studied different ways to estimate rainfall relationships using statistical methods called bivariate copulas. They found that one method, called the inference function of margins, works best for small datasets, while another adaptive maximization by parts, performs better with larger datasets.

Studies focused on specific regions also show how rainfall changes over time and impacts local conditions. Arjun [2017] looked at rainfall patterns in the drought-prone areas of Jalgaon District, Maharashtra He found big seasonal and yearly variations, which are important for farmers and water planners in the region. Another study byDzupire et al. [2020] used a statistical tool called the Frank copula to explore the connection between daily temperature and rainfall. This helped them understand how these two weather factors interact. Together, these studies show the importance of studying rainfall variability and its connections with other climate factors to tackle local challenges and plan better for the future.

# Data:

**Source: Indian Meterological Department Pune** by Arjun [2017]

## 3.3 Methodology

### 3.3.1 Scopes of the Study

In the simulation study, simulation data were generated from the Clayton copula as the true copula with four different values of the true copula parameter dependence that correspond to Kendall's tau values at $\tau = 0.20, 0.50, 0.60$, and 0.80. The sample sizes of the generated data are set at $n = 50, 100, 1000$, and 5000. Five hundred repetitions of the data generation and estimation process are performed for each combination of different sample size, $n$, and copula dependence level, $\theta$.

While, for the empirical study, rainfall data are used as the empirical data. The present study is based on the rainfall(mm) data collected from Indian mateorological Department, Pune for 30 years from 1980 to 2010, for 2 rain gauge stations,Socio-economic Review of Jalgaon district. The selected rain gauge station are Station Dharangaon and station Amalner were used and data taken from Arjun [2017].

The concepts of Copula 1.2, Maximum Likelihood Estimation 2.1.2, Inference Function for Margins 2.1.1 and the Adaptive maximization 2.1.5have been previously discussed and form the basis for the subsequent analysis.

### 3.3.2 Simulation Study

By Lokoman and Yusof [2018], It is difficult to estimate the copula dependence parameter, $\theta$, and to compare the three parametric estimation methods theoretically. Therefore, a simulation study was conducted in order to achieve the objectives. In the simulation study, simulation data are generated from the Clayton copula as the true copula with four different values of true copula parameter dependence that correspond to Kendall's tau, $\tau = 0.20, 0.50, 0.60$, and 0.80. The relationship of Kendall's tau ($\tau$) with the Clayton copula is shown in equation below.

$$\tau = \frac{\theta}{\theta + 2} \tag{3.1}$$

The sample sizes of the generated data are set to $n = 50, 100, 1000$, and 5000. Five hundred repetitions of data generation, estimation process, and squared error calculation are done for each combination of different data sample size, $n$, and copula dependence level, $\theta$. The performance of the three estimation methods and the estimators' precision were compared based on the measured root mean square error (RMSE). The RMSE formula is given as follows:

$$\text{RMSE}(\hat{\theta}) = \sqrt{\frac{1}{500} \sum_{i=1}^{500} \left(\hat{\theta}_i - \theta\right)^2} \tag{3.2}$$

where $\hat{\theta}_i$ is the estimator for the $i$th replication, and $\theta$ is the true parameter used in the simulation.

Table 3.1: The properties of Archimedean and Elliptical copulas

| Copula Family | Distribution Functions $C(u,v;\theta)$ | $\theta$ Range |
|---|---|---|
| Clayton | $(u^{-\theta}+v^{-\theta}-1)^{-1/\theta}$ | $\theta \geq -1$ |
| Ali-Mikhail-Haq | $\dfrac{uv}{1-\theta(1-u)(1-v)}$ | $\theta \in [-1,1]$ |
| Frank | $-\frac{1}{\theta}\ln\left[1+\frac{(e^{-\theta u}-1)(e^{-\theta v}-1)}{(e^{-\theta}-1)}\right]$ | $\theta \neq 0$ |
| Gumbel-Hougaard | $\exp\left[-\left((-\ln u)^{\theta}+(-\ln v)^{\theta}\right)\frac{1}{\theta}\right]$ | $\theta \geq 1$ |
| Gaussian | $\int\int\frac{1}{2\pi(1-\theta^2)^{1/2}}\phi^{-1}(v)\phi^{-1}(u)\exp\left\{-\frac{x^2-2xy\theta+y^2}{2(1-\theta^2)}\right\}dy\,dx$ | $\theta \in [-1,1]$ |
| Student's t | $\int\int\frac{1}{2\pi(1-\theta^2)^{1/2}}t^{-1}(v)t^{-1}(u)\left\{1+\frac{x^2-2xy\theta+y^2}{(1-\theta^2)}\right\}^{-(r+1)/2}dy\,dx$ | $\theta \in [-1,1]$ |

### 3.3.3 Empirical Study

In the empirical study, rainfall data were used in comparing the performance of the estimation methods. Two types of marginal distributions: Weibull and Gamma distributions are considered in fitting the hydrologic variables. This empirical study is limited only to the case of the bivariate copulas that are listed in Table 1.

The empirical study was conducted by the following procedures:

Step 1: Measure the dependency of the bivariate rainfall data to see the significance of the correlation and to check whether all the copula models listed in Table 1 can be used to model the dependency of the bivariate hydrologic data.

Step 2: Fit the bivariate hydrologic data with the choice of the marginal distributions through the goodness of fit test.

Step 3: Model the dependency of the bivariate hydrologic data by using the bivariate copulas that have been downsized from Step 1.

Step 4: Apply the three parameter estimation methods to estimate the copula dependency parameter, $\theta$.

Step 5: Assess the performance of the estimation methods and identify the best-fitted copula model through the goodness of fit test.

### 3.3.4 Goodness of Fit (GOF) Test

To select a fitted marginal distribution, the statistical goodness of fit (GOF) test was applied to the empirical study in this research. GOF test is a common method to verify the fitness of the statistical model to a set of observations. The best fitted marginal and copula distribution for this research were chosen based on the smallest value Akaike Information Criterion (AIC). The formula of AIC is written as:

$$\text{AIC} = 2p - 2\ln L \tag{3.3}$$

where $L$ is the value of the likelihood function based on the estimated parameters and $p$ is the number of estimated parameters in the statistical model.

As this study is mainly interested in the estimation of the copula dependence parameter $\theta$, the AIC values can be obtained by calculating the maximum likelihood of the copula log-likelihood model instead of using the full log-likelihood function . Therefore, for the copula goodness-of-fit (GOF) test, the formula for AIC can be expressed as:

$$\text{AIC} = 2p - 2\ln L \tag{3.4}$$

## 3.4 Results and Discussion

The estimation performance of ML estimation, IFM and AMBP methods were compared and evaluated in the simulation study based on the RMSE value. The three parametric estimation methods were then applied to the rainfall data in Station A and Station B to estimate the dependency between them.

### 3.4.1 Simulation Study

The root mean squared errors (RMSE) for the copula dependence parameter $\theta$, estimated by various methods, are presented in Table 3.2 for sample sizes $n = 50$, $n = 100$, $n = 1000$, and $n = 5000$. The rankings of each method, based on the measured RMSE, are illustrated in Figure 3.1, where Rank 1 indicates the method with the smallest RMSE, reflecting the best performance in parameter estimation. Notably, for the sample size $n = 50$, the Iterated Filtering Method (IFM) demonstrates superior precision, achieving the lowest RMSE across all correlation levels (Kendall's $\tau$ values of 0.20, 0.50, 0.60, and 0.80). In contrast, the ranks for the Alternating Multiplier Bootstrap (AMBP) and Maximum Likelihood Estimation (MLE) methods are inconsistent, although both yield similar RMSE values.

For sample size n = 100. IFM method shows higher precision with smaller RMSEs when $\tau = 0.20$, 0.50, and 0.60 but for $\tau = 0.80$, AMBP method has the smallest RMSE followed by MLE and IFM. For sample size $n = 1000$, it can be seen that AMBP has overtaken the ranking by showing the higher precision with small RMSEs for all correlation levels. The ranking is followed by MLE and IFM. Lastly, for sample size $n = 5000$. The results show that AMBP has the smallest RMSEs for all correlation levels. The ranking is followed by MLE and IFM.

Overall, the performance of the parametric estimation methods is different based on the sample size and the correlation level. When the sample is small, where $n = 50$, the Iterated Filtering Method (IFM) gives a more precise estimator than Maximum Likelihood Estimation (MLE) and the Alternating Multiplier Bootstrap (AMBP) for all correlation levels. For a sample size of $n = 100$, for $\tau = 0.2$, 0.5, and 0.6, IFM performs better than MLE and AMBP. However, when the correlation is very high, $\tau = 0.80$, AMBP and MLE methods provide more precise estimates than IFM. This is because IFM loses efficiency in estimation, as the first step of the IFM method only considers marginal parameters while disregarding the dependence level that may exist between the marginal random variables. For larger samples, $n = 1000$ and $n = 5000$, AMBP and MLE methods consistently yield more precise estimates than the IFM method across all correlation levels.

Therefore, based on the results of the simulation study, it can be said that for small sample size, $n < 1000$, the Iterated Filtering Method (IFM) estimator is more precise than the Alternating Multiplier Bootstrap (AMBP) and Maximum Likelihood Estimation (MLE) estimators for $\tau < 0.80$. However, for $\tau \geq 0.80$, the AMBP estimator is more precise than the MLE and IFM estimators. While for large sample sizes, $n \geq 1000$, the AMBP estimator is more precise than the MLE and IFM estimators for any correlation level.

The difference between the RMSE of AMBP and MLE estimators is very small since the AMBP estimator $\hat{\theta}_{\text{AMBP}}$ converged to the MLE estimator $\hat{\theta}_{\text{MLE}}$ as the iteration $k$ in Step $k$ of the AMBP algorithm tends to infinity. However, AMBP performs better than MLE because the AMBP estimator is updated until the smallest RMSE is computed, where $\hat{\theta}_{\text{AMBP}}$ converges to a constant value. Therefore, from the above results, it can be concluded that all the parametric methods could have the same performance when the sample size is large, although the correlation level is small.

Table 3.2: Comparison of the parametric estimation methods based on the RMSE of $\theta$ 5.2.1

| **Sample** | **Method** | $\tau = 0.2$ | | $\tau = 0.5$ | | $\tau = 0.6$ | | $\tau = 0.8$ | |
|---|---|---|---|---|---|---|---|---|---|
| **size,** $n$ | | RMSE | Rank | RMSE | Rank | RMSE | Rank | RMSE | Rank |
| 50 | MLE | 0.260940 | 3 | 0.568561 | 3 | 0.761583 | 2 | 1.804203 | 2 |
| | IFM | 0.251549 | 1 | 0.538741 | 1 | 0.725071 | 1 | 1.669725 | 1 |
| | AMBP | 0.260926 | 2 | 0.658558 | 2 | 0.761706 | 3 | 1.806007 | 3 |
| 100 | MLE | 0.182119 | 3 | 0.383016 | 3 | 0.511504 | 3 | 1.194508 | 2 |
| | IFM | 0.178910 | 1 | 0.379302 | 1 | 0.503107 | 1 | 1.208958 | 3 |
| | AMBP | 0.182053 | 2 | 0.382984 | 2 | 0.511315 | 2 | 1.189954 | 1 |
| 1000 | MLE | 0.051894 | 2 | 0.117036 | 2 | 0.152343 | 2 | 0.370964 | 2 |
| | IFM | 0.051885 | 1 | 0.116928 | 1 | 0.152218 | 1 | 0.376295 | 3 |
| | AMBP | 0.051893 | 3 | 0.116982 | 3 | 0.152318 | 3 | 0.370692 | 1 |
| 5000 | MLE | 0.024233 | 2 | 0.050937 | 2 | 0.075162 | 2 | 0.158973 | 2 |
| | IFM | 0.024243 | 3 | 0.051235 | 3 | 0.075954 | 3 | 0.160382 | 3 |
| | AMBP | 0.024215 | 1 | 0.050811 | 1 | 0.074592 | 1 | 0.158784 | 1 |

### 3.4.2 Empirical Study

In this section, the three parametric copula estimation methods were applied and compared for a joint distribution identification of the rainfall data. The rainfall data used in this study is selected from two rain gauge stations, Station Dharangaon, (Station A) and Station Amalner, (Station B). Their descriptive statistics are presented in Table 3.3

Table 3.3: Descriptive statistics of the daily rainfall for Station A and Station B 5.2.2

| **Descriptive statistics** | **Station A: Dharangaon** | **Station B: Amalner** |
|---|---|---|
| **Minimum (mm)** | 290.00 | 321.00 |
| **Maximum (mm)** | 1015.00 | 1640.00 |
| **Mean (mm)** | 624.93548 | 630.06452 |
| **Standard Deviation (mm)** | 173.63178 | 236.29740 |
| **Coefficient of Variation (CV)** | 27.78395% | 37.50368% |

The descriptive statistics for daily rainfall at Station A (Dharangaon) and Station B (Amalner) reveal notable differences in rainfall patterns. While the mean daily rainfall at both stations is quite similar—624.94 mm at Station A and 630.06 mm at Station B—the variability in rainfall is more pronounced at Station B. The minimum rainfall is 290.00 mm at Station A and 321.00 mm at Station B, with Station B having a much higher maximum of 1640.00 mm compared to Station A's 1015.00 mm. This indicates that Station B is more prone to extreme rainfall events.

Furthermore, the standard deviation, which measures the spread of rainfall values, is significantly higher at Station B (236.30 mm) than at Station A (173.63 mm), showing that

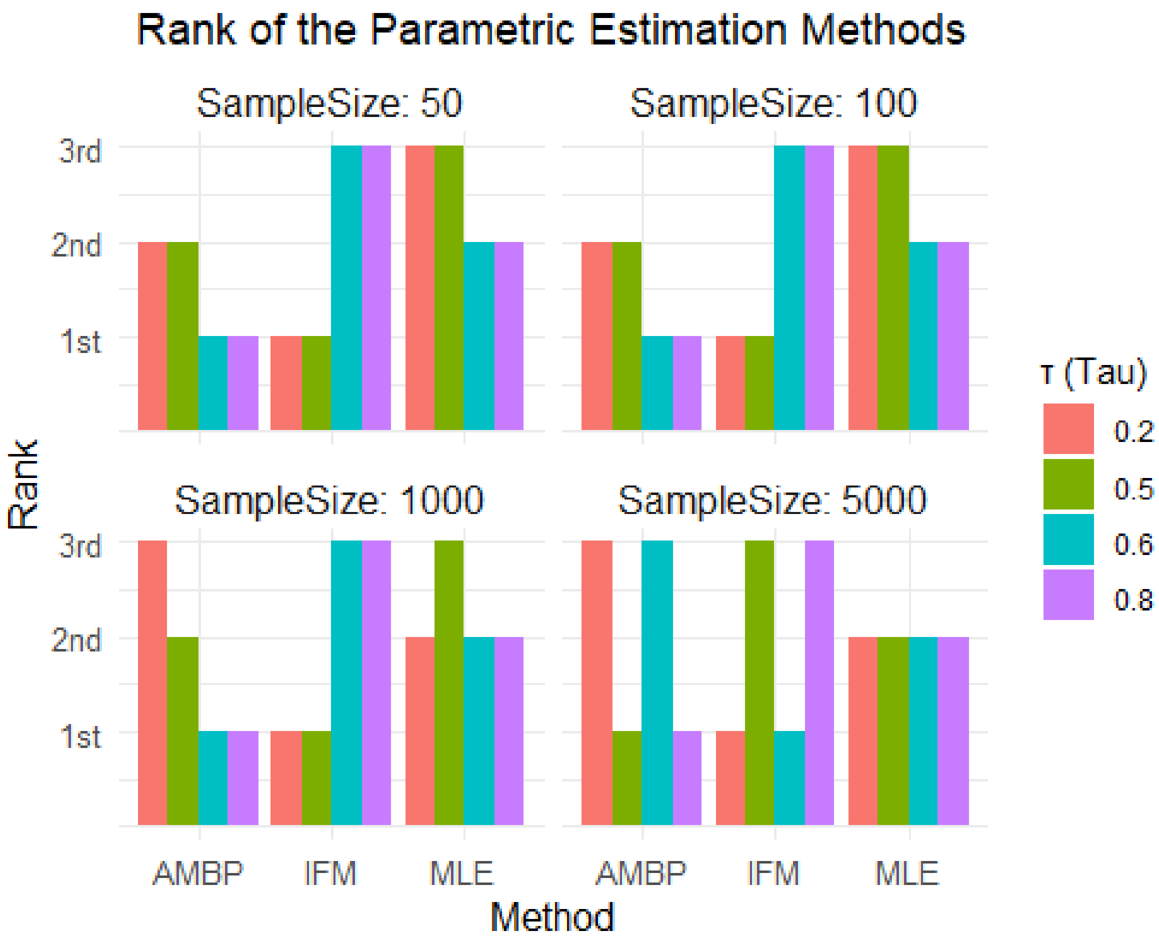

Figure 3.1: Comparison of the Parametric Estimation Methods

rainfall at Station B is more variable. The coefficient of variation (CV) further supports this, with Station B having a CV of 37.50% compared to 27.78% at Station A. A higher CV suggests that Station B's rainfall is less consistent and more unpredictable. In summary, although both stations have similar average rainfall amounts, Station B exhibits greater variability and is more likely to experience extreme fluctuations in daily rainfall.

### 3.4.3 The Correlation Level between the Rainfalls Data

The correlation between the rainfall data from Station A and Station B is shown in the scatter plot as follows.

In figure 3.2 observed that in the scatter plot, the rainfall data from Station A and Station B are positively correlated. The correlation between the two stations' rainfall data was measured using Kendall's tau method. The Kendall's tau value for the two series is 1, with a p-value approximately equal to 3.821e-15 at a significance level of $\alpha = 0.05$. Since the p-value is far less than 0.05, this indicates that the correlation for the rainfall data is statistically significant.

Given that the true copula and the copula dependence parameter ($\theta$) are unknown, the calculated Kendall's tau value can help narrow down the selection of suitable copulas. Based on the Kendall's tau value, several copulas from Table 3.1 are appropriate for modeling the dependence between Station A and Station B, including the Gumbel-Hougaard, Clayton,

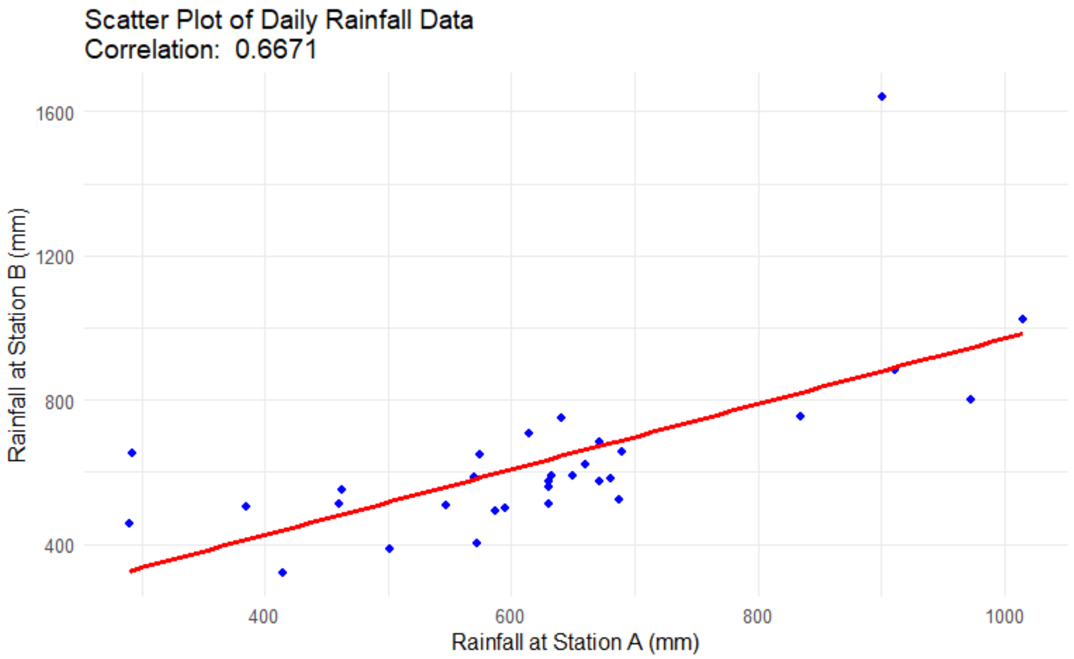

Figure 3.2: Scatter Plot of Daily Rainfall Data

Frank, Gaussian, and Student's t copulas. The Ali-Mikhail-Haq copula is not suitable because the Kendall's tau value of 1 falls outside its valid range ($\tau \in [-0.1817, 0.3333)$.

### 3.4.4 Marginal Distributions of the Daily Rainfall Data

In applying the copula parametric estimation methods to real hydrological data, the marginal distributions need to be identified first in order to avoid the misspecification of the marginal distributions. Two types of distributions were considered in fitting the daily rainfall data: Gamma and Weibull. In this study, the best-fitted marginal distributions were selected based on the goodness of fit test using the Akaike Information Criterion (AIC) measurement. The parameters of the fitted marginal distribution are estimated by using maximum likelihood estimation (MLE) which are shown in Table 3.4.

The Akaike Information Criterion (AIC) is used to evaluate the fit of two distributions—Gamma and Weibull. The AIC is a measure that helps identify the most suitable model, with lower values indicating a better fit for the data. For both stations, the Gamma distribution has the lowest AIC values (410.8326 for Station A and 417.343 for Station B), suggesting it is the best-fitting model. In comparison, the Weibull distribution has slightly higher AIC values (411.3758 for Station A and 429.0789 for Station B), indicating that while it may be a reasonable fit, it is not as effective as the Gamma distribution in capturing the characteristics of the rainfall data. Therefore, the results suggest that the Gamma distribution is the most appropriate choice for modeling the marginal distribution of daily rainfall data at both locations.

### 3.4.5 Joint Distribution of Daily Rainfall Data by Copula Method

The following copula estimation is then carried out for the daily rainfall data from the two stations. Gamma distribution is used as the marginal distributions for the parametric estimation methods: MLE, IFM, and AMBP since these methods need the marginal informa-

Table 3.4: Test of goodness-of-fit for marginal distribution based on the AIC result5.2.3.

| Marginal Distribution | AIC | |
|---|---|---|
| | Station A: Dharangaon | Station B: Amalner |
| Gamma | 410.8326 | 417.343 |
| Weibull | 411.3758 | 429.0789 |

tion.For Station A, the estimated shape parameter is $\alpha = 19.5117$ and the scale parameter is $\beta = 0.03043$. While, for Station B, the estimated shape parameter is $\alpha = 10.2729$ and the scale parameter is $\beta = 0.01630$. The relationship between rainfall at the two stations is yet to be determined, necessitating the identification of suitable copula models. In this study, after narrowing down the options, five copula candidates were selected to model the daily rainfall dependence at the stations. These include three Archimedean copulas: Gumbel-Hougaard, Clayton, and Frank, as well as two elliptical copulas: Gaussian and Student's t. The dependence parameters for these copulas were estimated using three methods, as shown in Table 5. Notably, the AMBP estimator closely matches the MLE estimator. This is because the AMBP algorithm keeps the estimators for $(\alpha_k, \beta_k, \theta_k)$ constant for each iteration, and as the number of iterations approaches infinity, it converges to the MLE of $(\alpha, \beta, \theta)$. This finding aligns with the studies by Lawless and Yilmaz [2011] and Kim et al. [2007] To select a fitted copula model and to measure the performance of estimation methods, the statistical goodness of fit (GOF) test has been applied for the empirical study in this research. The GOF test describes the fitness of the model to a set of observations. The best-fitted distribution is determined based on the minimum error produced, which is measured by Akaike Information Criterion (AIC) for this study. A small AIC value represents a better model fit. The AIC of each copula estimated by different estimation methods are listed in Table 3.6

Table 3.5: The estimators of the dependence parameter5.2.4.

| **Copula** | **MLE** | **IFM** | **AMBP** |
|---|---|---|---|
| Gumbel | 2.404434 | 1.827491 | 2.404434 |
| Clayton | 2.907166 | 2.034303 | 2.907166 |
| Frank | 8.959329 | 7.730989 | 8.959329 |
| Gaussian | 0.7708606 | 0.7708606 | 0.7708606 |
| Student's t | 0.7710453 | 0.8019733 | 0.7710453 |

Table 3.6: Test of goodness-of-fit for copula function based on the AIC result 5.2.5.

| **Copula** | **MLE** | **IFM** | **AMBP** |
|---|---|---|---|
| Gumbel | -24.184449 | -23.800000 | -24.184449 |
| Clayton | -3.812981 | -3.900000 | -3.812981 |
| Frank | -20.435430 | -19.800000 | -20.435430 |
| Gaussian | -21.549124 | -21.100000 | -21.549124 |
| Student's t | -19.915671 | -18.500000 | -19.915671 |

Table 3.6 shows that the AIC of the Gumbel copula estimated by MLE, IFM, and AMBP are smaller than the AIC of the other copulas. It shows that all estimation methods identify Table 3.6 shows that the AIC of the Gumbel estimated by MLE, IFM, and AMBP are smaller than the AIC of the other copulas. It shows that all estimation methods identify Gumbel copula

as the best one among the five candidate copulas that can describe the dependency of the rainfall data from Station A and Station B. Since the best-fitted copula has been determined, the performance of the three estimation methods can be compared based on the estimated copula estimator of Gumbel copula and the estimated AIC.

## 3.5 Conclusion

In analysis of daily rainfall data at Station A (Dharangaon) and Station B (Amalner) highlights significant differences in rainfall patterns, despite similar average rainfall amounts. Station B exhibits greater variability, with higher maximum rainfall, standard deviation, and coefficient of variation, indicating a greater susceptibility to extreme rainfall events. The Gamma distribution was identified as the most appropriate model for the marginal distribution of daily rainfall at both stations, as it demonstrated the lowest AIC values compared to other distributions. Furthermore, the Gumbel copula was found to be the best model for capturing the dependency between the rainfall patterns at the two stations, with consistent performance across multiple estimation methods (MLE, IFM, and AMBP). These findings suggest that Station B's rainfall is less predictable and more extreme, and that the Gamma distribution, along with the Gumbel copula, provides the most reliable framework for modeling the daily rainfall data and their dependencies.

# Chapter 4

# Application of Copulas in Multivariate Process Control Charts

## Abstract

This study explores the application of copulas in the development of multivariate process control charts, highlighting their advantages over traditional control charts. Traditional control charts, such as Shewhart and Hotelling's T-squared charts, assume normality and independence among variables, limiting their ability to capture complex dependencies and tail behaviours in multivariate data. By utilizing the Copper Wire Production Line Dataset Osroru [2020], this research demonstrates how copula-based control charts can model non-linear dependencies, capture tail dependencies, and provide a comprehensive understanding of the joint behaviour of multiple variables. The study outlines the construction of copula-based control charts, including Hotelling's T-squared and MCUSUM charts, emphasizing their flexibility in accommodating different marginal distributions and detecting small shifts in the process mean. The findings suggest that copula-based control charts offer enhanced sensitivity and accuracy in monitoring multivariate processes, particularly in industries where quality control is critical.

## 4.1 Multivariate Control Charts: An Overview

**As we have discussed the concept of copulas and Sklar's Theorem in Section 1.1,** copulas serve as powerful tools for modeling dependencies in multivariate processes, allowing flexibility in handling non-linear relationships and deviations from normality. This approach transforms variables to a uniform scale $[0,1]$ using their marginal CDFs, enabling flexible dependency modelling Krupskii et al. [2020].

Traditional univariate control charts may fail to detect shifts in a process when multiple quality characteristics are correlated. Explicit copulas, such as the Gaussian, Student's t, Clayton, and Gumbel copulas, allow for the modeling of the joint distribution of these characteristics. Copula families are chosen based on observed data characteristics, such as tail dependencies, making them particularly useful for constructing advanced control charts that accurately monitor real-world processes with complex dependencies.

### 4.1.1 Hotelling's T-Squared Statistic

Hotelling's T-squared statistic is a generalization of the Student's t-test to multiple variables, widely used in multivariate statistical analysis to detect if the sample means of several correlated variables significantly differ from a reference mean vector. It measures the distance between a sample mean vector and a reference mean vector, accounting for the variance-covariance structure of the variables discussed in their paper Verdier [2013].

- **Mathematical Definition:** For a sample of $p$ variables and $n$ observations, Hotelling's T-squared statistic is computed as:

  $$T^2 = n\left(\bar{X}-\mu\right)'\Sigma^{-1}\left(\bar{X}-\mu\right)$$

  Where:

  - $\bar{X}$: Sample mean vector.
  - $\mu$: Reference mean vector.
  - $\Sigma^{-1}$: Inverse of the covariance matrix.
  - $n$: Number of observations.

- **Utility in Multivariate Control Charts:**

  - Traditional univariate control charts, such as the Shewhart chart, monitor individual variables independently, which may fail to detect shifts in processes with correlated variables.
  - Hotelling's T-squared statistic considers relationships among variables, summarizing deviations of multiple quality characteristics into a single statistic to detect multivariate out-of-control signals.
  - Particularly useful for capturing process variations involving multiple, dependent variables by Lestari et al. [2019].

- **Control Limits:** The statistic for each observation is compared to a control limit derived from the chi-squared or F-distribution based on a desired confidence level (e.g., 95% or 99%):

  $$T^2 \text{ control limit} = \chi^2_{0.95,p}$$

  where $p$ is the number of variables.

- **Process Shift Detection:** Hotelling's T-squared statistic captures shifts in the mean vector while accounting for correlation among variables, making it sensitive to both large and small shifts in multivariate processes.

- **Extensions Using Copulas:** Incorporating copulas allows for capturing non-linear relationships among variables, improving the robustness of T-squared control charts for complex dependency structures.

### 4.1.2 MCUSUM Chart

The Multivariate Cumulative Sum (MCUSUM) control chart is an extension of the univariate CUSUM chart, designed to monitor multiple correlated variables. It is particularly effective for detecting small, sustained shifts in the process mean, offering greater sensitivity than traditional multivariate charts introduce in Busababodhin and Amphanthong [2016].

- **CUSUM Statistic for Multivariate Data:**
  - The MCUSUM statistic is based on the Mahalanobis distance, which accounts for the covariance structure of the variables while measuring the distance of the process observation from the target mean.
  - The CUSUM vector $C_t$ is computed as:

    $$C_t = \max\left(0, C_{t-1} + \Sigma^{-1}\left(X_t - \mu\right)\right)$$

    Where:

    - * $X_t$: Multivariate observation vector at time $t$,
    - * $\mu$: In-control mean vector,
    - * $\Sigma$: Covariance matrix of the multivariate data.

- **Advantages of MCUSUM Charts:**
  - **Sensitivity to Small Shifts:** Quickly detects small, sustained shifts in the process mean.
  - **Capturing Dependencies:** Incorporates copulas to model non-linear dependencies, resulting in a more robust representation of process relationships.
  - **Comprehensive Monitoring:** Aggregates information from multiple variables into a single statistic, simplifying multivariate process control resulted asCrosier [1988].

### 4.1.3 Joint Density-Based Control Chart

A joint density-based control chart models dependencies between process variables, making it particularly effective in scenarios where extreme values co-occur more frequently. By utilizing a suitable copula, the joint probability density of the variables is calculated, which enables more accurate monitoring of their collective behavior introducced in Fatahi et al. [2011]. Control limits are set based on statistical thresholds derived from this joint density, allowing for precise detection of shifts or outliers in the multivariate process.

**Advantages of Joint Density-Based Control Charts:**

- Captures non-linear dependencies and tail dependence effectively.
- Models a wider range of dependence structures compared to traditional methods.
- Identifies outliers and shifts in multivariate processes with high precision.

## 4.2 Constructing Copula-Based Control Charts

By Ahmad et al. [2024] introducess the process for constructing copula-based control charts involves the following steps:

- **Assess Dependencies Among Variables:**
  - Use tools such as correlation plots, scatter plots, or Spearman's rank correlation coefficient to evaluate dependencies among variables.
  - Visualize relationships to identify patterns or correlations.
- **Determine Marginal Distributions:**
  - Fit various distributions (e.g., Normal, Exponential, Weibull) to individual variables in the dataset.
  - Use the Kolmogorov-Smirnov (KS) test to identify the best-fit distribution for each variable.
- **Standardize Variables:**
  - Transform each variable to a uniform [0,1] scale using its cumulative distribution function (CDF).
  - This step ensures the variables are standardized for copula modeling.
- **Fit a Copula Model:**
  - Select an appropriate copula model (e.g., Gumbel, Gaussian, Clayton) based on the nature of dependencies observed in Berg and Bakken [2007]
  - Fit the copula to the standardized data using methods such as maximum likelihood estimation.
- **Evaluate Copula Fit:**
  - Assess the goodness-of-fit of the copula model using tools like the KS statistic or other appropriate tests.
  - Choose the copula model with the best fit for further analysis and control chart construction.

## 4.3 Construction of Different Types of Copula-Based Multivariate Control Charts

1. **Control Chart Based on Joint Probability Density**

   Method Overview: This method monitors the joint probability density (PDF) of the observations using the fitted copula. If the joint density falls below a specified threshold, the process is flagged as out-of-control.

   (a) Transform each variable into the uniform [0,1] scale using its marginal CDF.

(b) Fit the appropriate copula (i.e., the copula that is fitted to the data) to the transformed data.

(c) Compute the joint density for each observation using the copula's PDF.

(d) Set control limits based on the 5th percentile of joint density values. Any point below this threshold is flagged as out-of-control.

(e) Plot the control chart with the joint density values on the y-axis and observation indices on the x-axis.

2. **Multivariate Cumulative Sum (CUSUM) Chart**

   Method Overview: The MCUSUM chart monitors cumulative deviations from expected joint behavior, making it sensitive to small shifts in the process.

   **Steps:**

   (a) Transform each variable to the uniform space using its marginal CDF.

   (b) Fit the appropriate copula (i.e., the copula that is fitted to the data) to the transformed data.

   (c) Compute the joint CDF from the fitted copula.

   (d) Calculate the CUSUM statistic by summing deviations from the mean of the joint CDF.

   (e) Plot the MCUSUM chart, tracking cumulative deviations over time.

3. **Copula-Based Hotelling's T-Squared Control Chart**

   Method Overview: This method applies Hotelling's T-squared statistic to the copula-transformed data, combining multivariate control with dependency modeling.

   (a) Fit the appropriate copula (i.e., the copula that is fitted to the data) to the transformed data.

   (b) Transform the data into a uniform scale using the fitted copula.

   (c) Compute Hotelling's T-squared statistic for the copula-transformed data:

   $$T^2 = n\left(\bar{X} - \mu\right)' \Sigma^{-1} \left(\bar{X} - \mu\right)$$

   where $\Sigma$ is the covariance matrix of the transformed data.

   (d) Set control limits based on the chi-squared distribution with $p$ degrees of freedom.

   (e) Plot the Hotelling's T-squared control chart to monitor process stability.

## 4.4 Practical Application of Copula-Based Multivariate Control Charts

1. **Dataset Overview** The Copper Wire Production Line dataset on Kaggle is designed for root cause analysis in a real copper wire production line dataset taken from Osroru [2020]. Below are its key details:

   - Author: The dataset is provided by a user named osroru on Kaggle.

- Variables:
  - *Machine:* Identifier for the machine used in production.
  - *Shift:* The shift during which the data was recorded (e.g., A, B, C).
  - *Operator:* Identifier for the operator.
  - *Date:* The date when the data was recorded.
  - *Cable Failures:* Number of cable failures.
  - *Cable Failure Downtime:* Downtime in minutes due to cable failures.
  - *Other Failures:* Number of other types of failures.
  - *Other Failure Downtime:* Downtime in minutes due to other failures.
- Units:
  - *Cable Failures* and *Other Failures* are counts (unitless).
  - *Cable Failure Downtime* and *Other Failure Downtime* are measured in minutes.
- **Purpose:** The dataset is intended for analyzing and identifying the root causes of failures in the production line, which can help improve efficiency and reduce downtime.

2. **Correlation Analysis** The dependency between the variables in the dataset is analyzed using Spearman's rho and Kendall's tau correlation coefficients explain in Sukparungsee et al. [2017]. The results are presented below:

Table 4.1: Spearman's Rho Correlation Matrix

| | **CF** | **CF Downtime** | **OF** | **OF Downtime** |
|---|---|---|---|---|
| **Cable Failures** | 1.000 | 0.875 | -0.445 | -0.454 |
| **Cable Failure Downtime** | 0.875 | 1.000 | -0.464 | -0.457 |
| **Other Failures** | -0.445 | -0.464 | 1.000 | 0.964 |
| **Other Failure Downtime** | -0.454 | -0.457 | 0.964 | 1.000 |

Cable Failures : CF, Other Failures : OF

Table 4.2: Kendall's Tau Correlation Matrix

| | **CF** | **CF Downtime** | **OF** | **OF Downtime** |
|---|---|---|---|---|
| **Cable Failures** | 1.000 | 0.768 | -0.390 | -0.372 |
| **Cable Failure Downtime** | 0.768 | 1.000 | -0.380 | -0.356 |
| **Other Failures** | -0.390 | -0.380 | 1.000 | 0.884 |
| **Other Failure Downtime** | -0.372 | -0.356 | 0.884 | 1.000 |

3. **Fitted Distributions** To model the variables in the dataset, several distributions were fitted, and the goodness-of-fit was evaluated using the KS statistic and p-value. The results are summarized in the table below:

   The best-fitting distributions for each variable are as follows:

   - **Cable Failures:** t distribution with parameters (5.93, 1.24, 1.09)
   - **Cable Failure Downtime:** Normal distribution with parameters (51.44, 56.47)

Table 4.3: Fitted Distributions and Goodness-of-Fit

| Variable | Distribution | KS Statistic | p-value |
|---|---|---|---|
| Cable Failures | t | 0.214 | $1.79 \times 10^{-6}$ |
| Cable Failure Downtime | norm | 0.181 | $9.47 \times 10^{-5}$ |
| Other Failures | norm | 0.315 | $1.26 \times 10^{-13}$ |
| Other Failure Downtime | norm | 0.329 | $7.78 \times 10^{-15}$ |

- **Other Failures:** Normal distribution with parameters (0.54, 0.97)
- **Other Failure Downtime:** Normal distribution with parameters (47.05, 106.04)

4. **Marginal Distributions** The marginal distributions for the variables were fitted and visualized to confirm the appropriateness of the chosen distributions. The plot below illustrates the marginal distributions for each variable in the dataset.

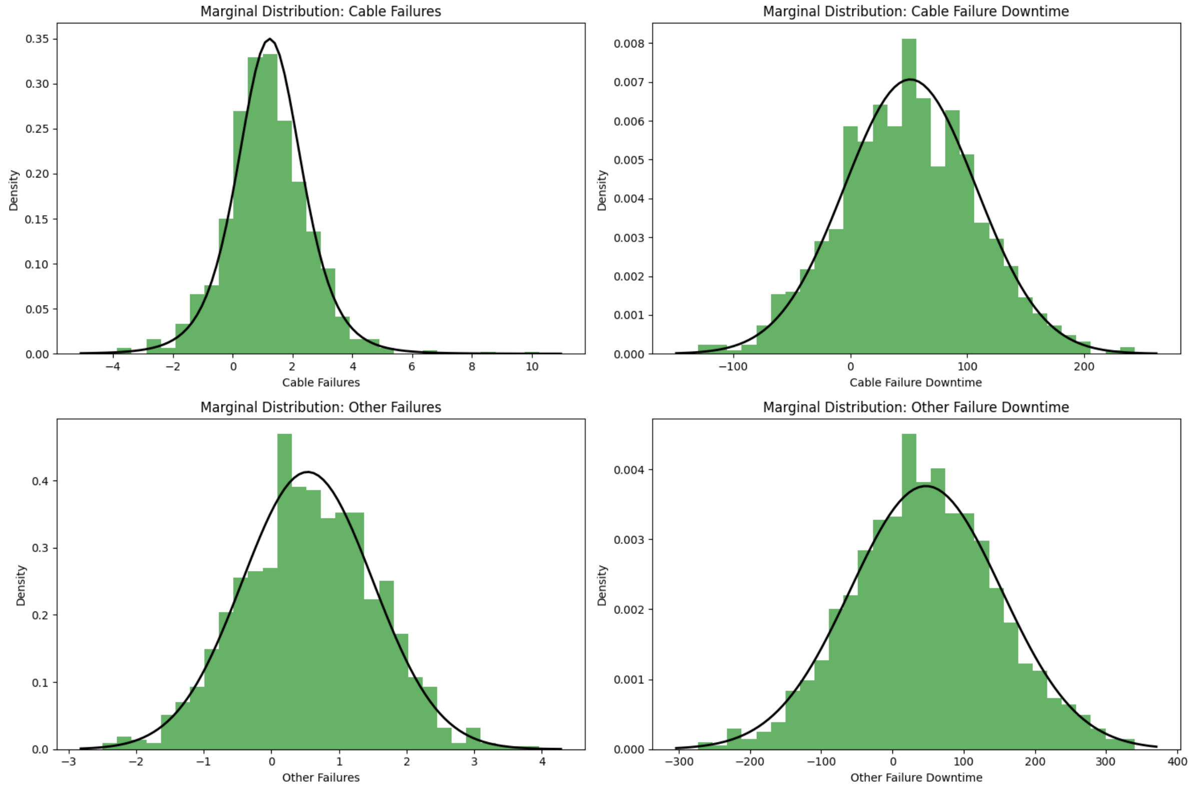

Figure 4.1: Marginal Distributions Plot

5. **Modeling Dependencies with Copulas** To model the dependencies between the variables, the copulas library was used to fit a Gaussian copula model more study refer Yan [2007]. The steps involved are as follows:

   (a) **Fit a Copula:** Choose and fit a copula model to the marginal CDFs.

   (b) **Estimate Parameters:** Extract and examine the copula parameters (e.g., covariance matrix for Gaussian, $\theta$ for Clayton).

   (c) **Generate Joint Samples:** Use the fitted copula to generate joint samples, modeling the joint distribution of the original variables.

The results for the fitted copula models are summarized in the table below:

Table 4.4: Fit Results for Each Copula

| **Copula** | **Parameters** |
|---|---|
| Clayton Copula | 0.6859 |
| Gumbel Copula | 1.0757 |
| Student Copula | `df=4.462,` $\rho$`[0.890, -0.568, -0.565, -0.556, -0.543, 0.962]` |
| T Copula | `df=4.462,` $\rho$`[0.890, -0.568, -0.565, -0.556, -0.543, 0.962]` |
| Gaussian Copula | `[0.884, -0.535, -0.538, -0.550, -0.538, 0.937]` |

6. **Goodness-of-Fit Test and Best Fitted Copula** The scatter plot of simulated values from the fitted copulas, as shown in the figure 4.2 below, demonstrates the simulated data for each copula model. To determine the best fit, we performed the Kolmogorov-Smirnov (KS) test, a goodness-of-fit test in Berg and Bakken [2007], for each copula. The KS statistic and associated p-value for each copula are as follows:

Table 4.5: Goodness-of-Fit Test Results for Each Copula

| **Copula** | **KS Statistic** | **P-Value** |
|---|---|---|
| Clayton Copula | 0.215 | 0.0020 |
| Gumbel Copula | 0.208 | 0.0031 |
| Student Copula | 0.235 | 0.0005 |
| T Copula | 0.248 | 0.0002 |
| Gaussian Copula | 0.248 | 0.0002 |

Based on the Kolmogorov-Smirnov test results, the best fit copula is the Gumbel Copula, as it has the smallest KS statistic and the highest p-value compared to the other copulas.

**Best Fitted Copula: Gumbel Copula**
The parameter for the Gumbel Copula is:

$$\theta = 1.0757$$

**Interpretation:** The Gumbel Copula with a parameter $\theta \approx 1.076$ indicates moderate upper tail dependence. This suggests that the Gumbel copula is well-suited for modeling the joint behavior of variables where extreme values tend to occur together in the upper tail of the distribution.

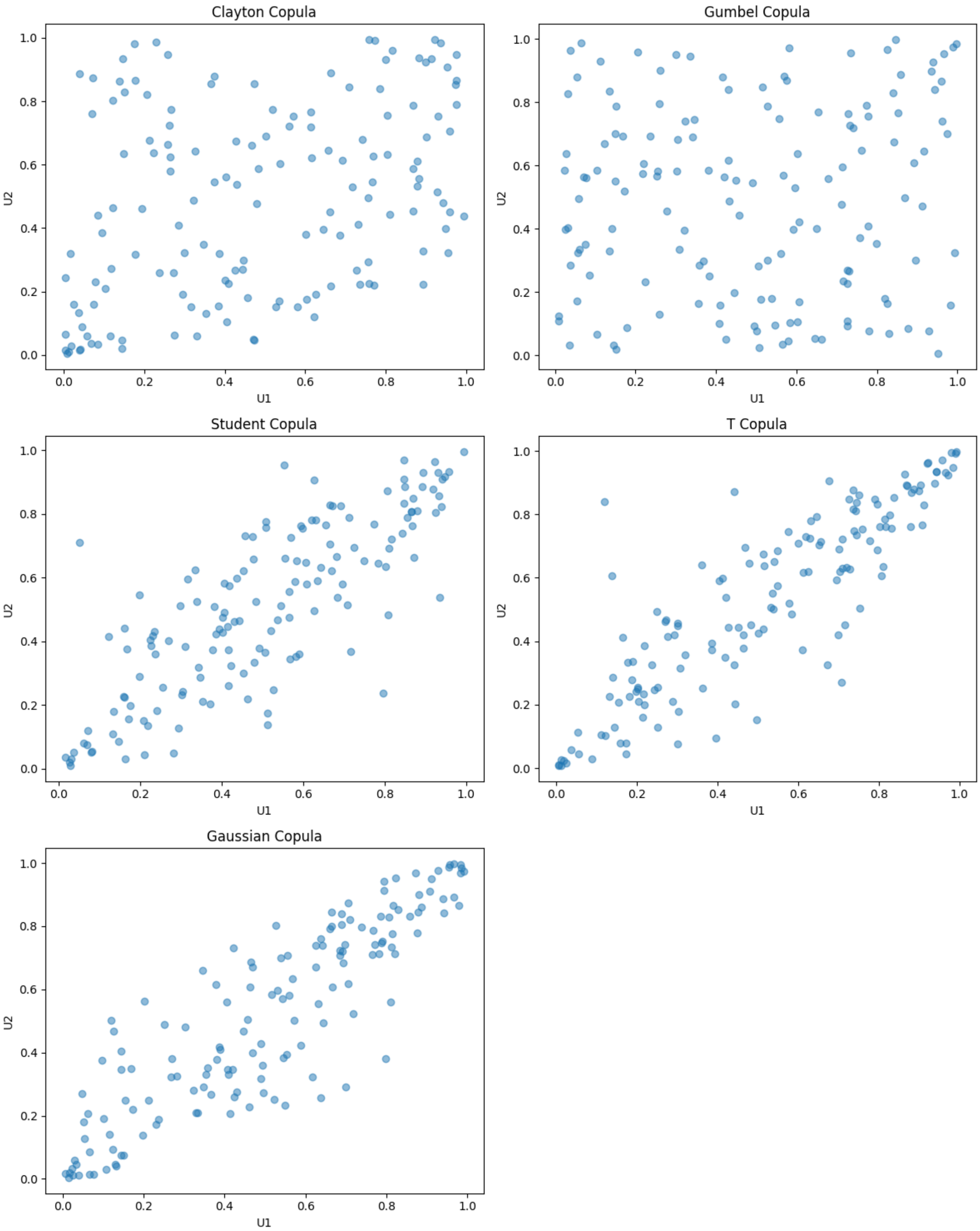

Figure 4.2: The Scatter-plot of Simulated Values from the Fitted Copulas

**Conclusion:** Based on the KS test and the parameter values, the Gumbel Copula is the best model for capturing the dependencies between the variables in this analysis.

When implementing multivariate control charts based on copulas, there are several methods you can use to monitor the joint behavior of multiple variables. There are various ways to plot the control chart based on a fitted copula. Here are some approaches:

7. **Copula-Based Hotelling's T-Squared Control Chart:** This control chart is widely used for multivariate process monitoring. By leveraging copulas, you can first transform the data into a uniform scale and then apply Hotelling's T-squared method to the transformed data, as we have discussed earlier in 3.

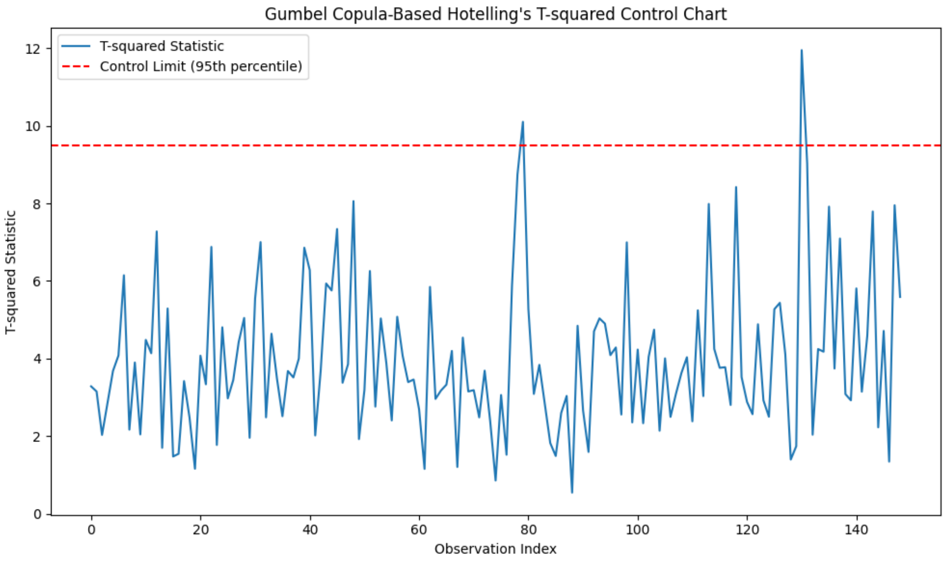

Figure 4.3: Gumbel Copula-Based Hotelling's T-squared Control Chart

There are several methods to construct copula-based control charts, but here the Gumbel copula is fitted to our data. Considering the structure of the data, Hotelling's T-squared control chart is a solid option for monitoring your multivariate process. This method is particularly effective for identifying shifts in the mean of a multivariate process and can be adapted to work with copula-transformed data.

Once you've transformed the data using the Gumbel Copula, the chart will monitor deviations from the mean for the joint behavior of the variables. This is particularly useful in quality control scenarios where you're monitoring for shifts in the joint distribution of multiple correlated variables.

If the focus is on detecting tail dependencies or extreme values, which the Gumbel Copula is good at capturing, the multivariate cumulative sum (CUSUM) chart or joint probability density chart might also be useful. These methods emphasize the extremes and cumulative behavior, which could complement the use of Hotelling's T-squared for capturing overall shifts in the joint process.

8. **Multivariate Cumulative Sum (CUSUM) Chart:** The MCUSUM chart balances sensitivity and specificity by calculating expected joint behavior, observing and expected values, and setting control limits using the chi-square distribution. The procedure to construct this chart is already discussed in 2

   Using the absolute values of CUSUM when setting control limits is beneficial for several reasons:

   - **Symmetry:** It allows for symmetric control limits around the target, capturing deviations in either direction from the norm.

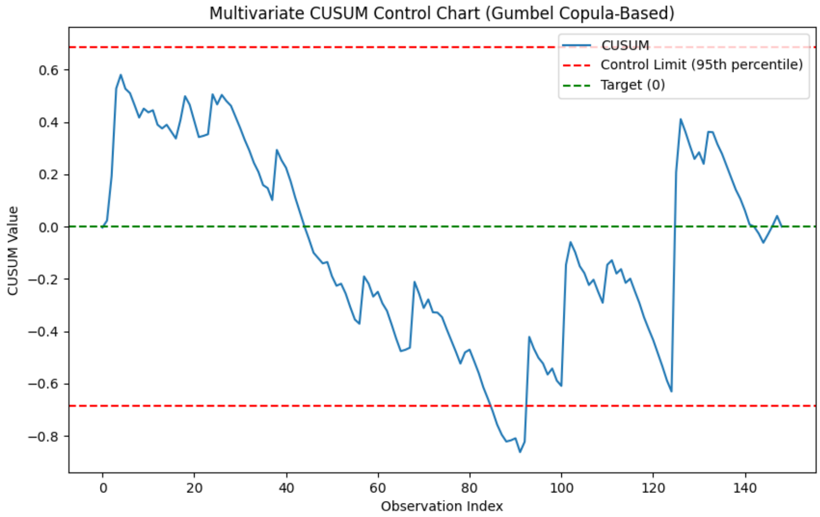

Figure 4.4: Multivariate CUSUM Control Chart (Gumbel Copula-Based)

- **Sensitivity:** This method increases sensitivity to any significant shifts, regardless of whether these shifts are above or below the target mean.
- **Simplification:** Simplifies monitoring by using a single decision rule—if the absolute CUSUM exceeds the threshold, it signals an out-of-control condition.

This approach ensures comprehensive monitoring of all significant deviations, enhancing the effectiveness of quality control efforts.

9. **Control Chart Based on Joint Probability Density** The process involves fitting a copula to data, calculating joint probability density for each observation, setting control limits, and transforming variables into uniform space. The Gumbel copula is then fitted, and joint density is computed for each observation. Control limits are set based on joint density values.The detailed steps to construct this chart are explained in 1.

   **Note:** Tail Dependency: The Gumbel copula models upper tail dependency strongly. If there are observations in your data that are extreme relative to others, the Gumbel copula's ability to emphasize these upper tail dependencies can result in a high joint density value for such points. This is especially true if these extreme observations are more aligned (i.e., they show a tendency to occur together), which is what the Gumbel copula captures.

## 4.5 Advantages of Using Gumbel Copula in Control Charts

1. **Enhanced Sensitivity:** This method enhances sensitivity to detect changes, especially in the tails of the distribution, which are critical in risk management and quality control scenarios.

2. **Comprehensive Monitoring:** It provides a comprehensive view by considering the interdependencies between multiple variables, which might be overlooked in other types of charts.

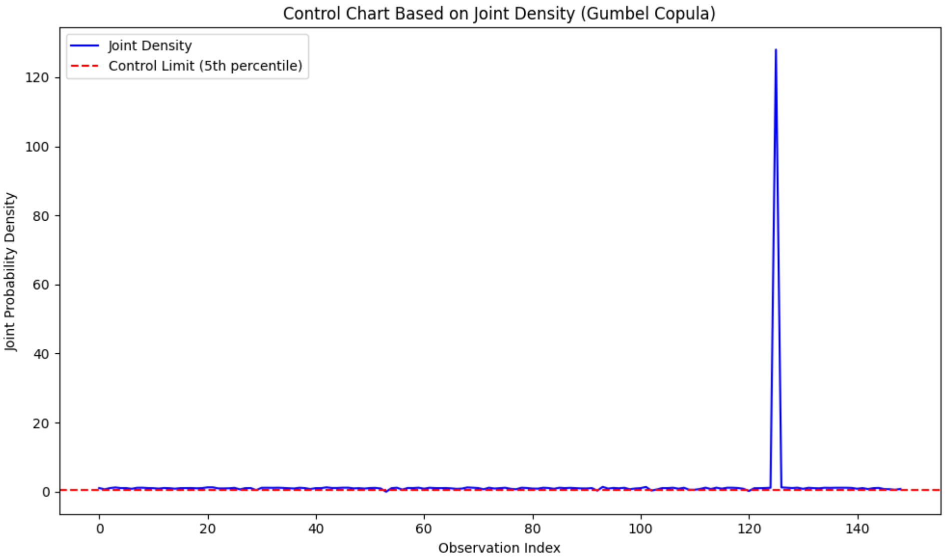

Figure 4.5: Control Chart Based on Joint Density (Gumbel Copula)

3. **Flexibility in Modelling Dependencies:** Unlike some copulas, the Gumbel copula can model right tail dependencies, which are useful in many real-world applications where extreme values of one variable are likely to be associated with extreme values of another.

4. **Outlier Detection:** By examining the joint probability density, this type of control chart can effectively identify outliers or unusual combinations of variable values. These outliers might indicate anomalies, errors, or process issues.

5. **Captures Positive Dependence:** The Gumbel copula is particularly well-suited for modelling positive dependence between variables. This is often relevant in real-world scenarios where increases in one variable tend to be associated with increases in others.

6. **Tail Dependence:** The Gumbel copula can capture tail dependence, which is important in situations where extreme values of one variable are associated with extreme values of another.

## 4.6 Comparative Analysis of Copula-Based Control Charts

| **Feature** | **Hotelling's T-squared** | **MCUSUM** | **Joint Density-Based** |
|---|---|---|---|
| Sensitivity | Medium to high; better for large shifts | High; excels at detecting small shifts | Very high; sensitive to extremes and subtleties in joint distributions |
| Best for | Detecting significant shifts in the mean of multivariate data | Detecting small, continuous changes in the process | Monitoring the overall distribution, particularly extremes |

| Feature | **Hotelling's T-squared** | **MCUSUM** | **Joint Density-Based** |
|---|---|---|---|
| Complexity | Moderate; involves matrix calculations | High; requires cumulative sum calculations and continuous updating | High; involves understanding complex density functions and tail dependence |
| Data Requirements | Relatively flexible; less data needed for effective monitoring | Requires a consistent stream of data to track changes accurately | Requires high-quality data to accurately estimate densities |
| Interpretation | More straightforward as it deals with shifts in mean values | More complex due to the accumulation of deviation values | Complex; requires statistical expertise to interpret density functions correctly |
| Detection Speed | Quick response to changes but slower than MCUSUM in detecting small shifts | Fast; accumulates deviations quickly, providing quicker detection of trends | Speed depends on the settings of density thresholds; can be tuned for rapid detection |
| Tail Dependency | Limited capability to handle tail dependency | Better than Hotelling's but not specifically designed for tail dependencies | Specifically designed to handle tail dependencies efficiently; ideal for data with significant tail-driven risks |
| Setup Difficulty | Moderate; easier than MCUSUM but requires understanding of multivariate statistics | High; requires detailed setup of control limits based on statistical distributions | High; demands in-depth knowledge of copulas and joint distribution modeling |
| Flexibility | Limited to linear relationships between variables | Good; can be adapted to various process settings but is less flexible than joint density-based approaches | Excellent; highly adaptable to various types of data interdependencies |
| Pros | Good for quick detection of major shifts, straightforward setup and analysis | Highly sensitive to small shifts, making it suitable for critical quality control processes | Exceptional at modeling non-linear dependencies, providing a nuanced view of multivariate relationships |
| Cons | Less effective at detecting subtle changes and handling complex dependency structures | Can be too sensitive, leading to false alarms without proper calibration | Complex to set up and interpret; requires substantial statistical training to utilize effectively |
| Typical Applications | Suitable for quality control in less complex multivariate settings | Ideal for high-tech and sensitive manufacturing processes where early detection of small anomalies is crucial | Best for financial and risk management sectors where understanding joint risk and tail dependencies is crucial |

| **Feature** | **Hotelling's T-squared** | **MCUSUM** | **Joint Density-Based** |
|---|---|---|---|
| Control Limit Calculation | Based on the chi-square distribution with 95% confidence intervals | Based on the chi-square distribution, reflecting the statistical distribution of the CUSUM statistic under the null hypothesis that the process remains in control | Based on threshold percentiles (typically 5%) of the density estimates |
| Monitoring Focus | Focuses on mean shifts in a multivariate normal framework | Focuses on accumulative deviations, enhancing sensitivity to prolonged minor deviations | Focuses on the entire distribution, especially the behaviour in the tails, making it robust against rare but critical events |

Table 4.6: Comparative Analysis of Copula-Based Control Charts

## 4.7 Traditional Multivariate Control Charts

In this section, we explore traditional multivariate control charts that do not incorporate copula-based methodologies. These charts provide effective tools for monitoring the joint behavior of multiple variables in a process.

1. **Hotelling's T-squared Control Chart:** The Hotelling's T-squared control chart is a multivariate extension of traditional control charts. It monitors the joint behavior of multiple variables by calculating a statistic that captures shifts in the mean of the process introduce inVerdier [2013]. This method is particularly useful in detecting significant deviations from the expected mean vector in a multivariate framework.

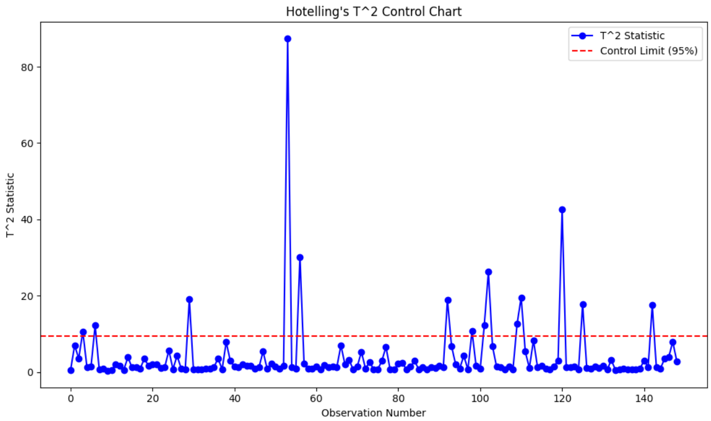

Figure 4.6: Hotelling's T-squared Control Chart

2. **Multivariate CUSUM (MCUSUM) Control Chart:** The multivariate CUSUM (MCUSUM) control chart tracks cumulative deviations from the process mean over time. Unlike Hotelling's T-squared chart, it gives more weight to recent changes, making it highly effective in detecting small, continuous shifts in the processSukparungsee et al. [2017]. The MCUSUM is widely used in quality control scenarios where early detection of gradual shifts is critical.

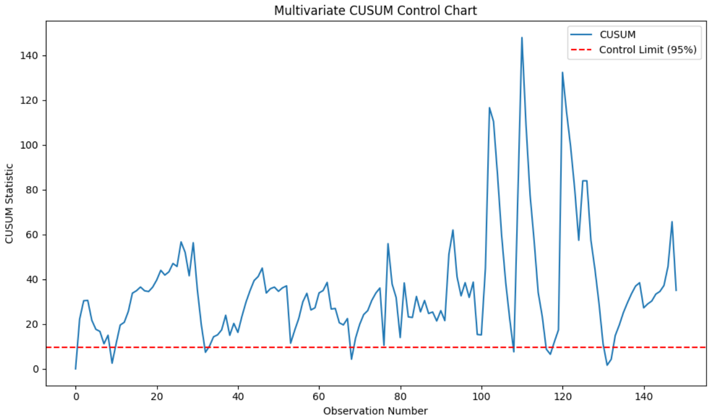

Figure 4.7: Multivariate CUSUM Control Chart

## 4.8 Comparison: Traditional vs Copula-Based Control Charts

| **Aspect** | **Traditional Control Charts** | **Copula-Based Control Charts** |
|---|---|---|
| Use Case | Best for univariate or multivariate independent data. | Best for multivariate data with complex dependencies (non-linear, tail dependence). |
| Dependency Modelling | Assumes independence or linear correlation between variables (especially for multivariate). | Models complex dependencies, including non-linear relationships and tail dependencies. |
| Marginal Distribution Assumption | Assumes variables follow normal distributions (or other specific distributions). | Allows flexibility with marginal distributions (e.g., normal, t-distribution, etc.). |
| Handling of Joint Behaviour | Primarily focuses on individual variables or simple aggregated statistics (e.g., multivariate T-squared). | Directly models the joint behaviour of variables through a copula, accounting for dependencies across all values. |
| Tail Dependence | Limited ability to detect extreme joint values or tail dependencies between variables. | Captures tail dependencies (e.g., Gumbel copula for upper tail dependencies), crucial in identifying joint extremes. |
| Control Limits | Control limits are based on individual distributions or simple aggregate metrics. | Control limits are based on the joint probability density or conditional probabilities. |
| Sensitivity to Shifts | Can miss subtle shifts in the joint behaviour of variables (e.g., if individual variables remain in control but their interaction changes). | More sensitive to subtle joint shifts in multiple variables, especially when variables interact. |

*Continued on next page...*

| **Aspect** | **Traditional Control Charts** | **Copula-Based Control Charts** |
|---|---|---|
| Ease of Implementation | Simple and widely used, especially for univariate monitoring or independent variables. | More complex to implement due to copula fitting, but provides deeper insights for multivariate processes. |
| Multivariate Extensions | Available (e.g., Hotelling's T-squared), but assumes normality and focuses on mean shifts. | Captures complex multivariate relationships; can handle both mean shifts and dependency shifts. |
| Non-Linear Relationships | Struggles with non-linear dependencies between variables. | Can model and monitor non-linear dependencies effectively. |
| Handling of Extreme Events | May overlook joint extreme events across variables. | Suitable for detecting joint extreme events (tail dependencies), especially important in risk or quality management. |
| Flexibility | Less flexible; relies on strong assumptions about the data (e.g., normality). | Highly flexible; can accommodate different distributions and complex dependency structures. |
| Computational Complexity | Computationally simple and fast to implement. | More computationally intensive due to copula fitting, but necessary for multivariate processes. |
| Examples of Use | Monitoring one variable at a time or simple multivariate processes (e.g., independent variables). | Complex quality control processes (e.g., manufacturing where multiple quality metrics are correlated). |

Table 4.7: Comparison of Traditional and Copula-Based Control Charts

## 4.9 Conclusion

The comparison highlights the significant advantages of copula-based control charts over traditional control charts in handling real-world quality control challenges. While traditional charts are effective for univariate data and assume linear dependencies, they fall short when dealing with complex, non-linear relationships or datasets that deviate from normality.

In contrast, copula-based control charts, like those constructed using the Gumbel Copula in our study, excel at capturing joint dependencies, including tail dependencies, and modelling non-linear interactions between variables. Their ability to accommodate diverse marginal distributions and monitor joint probabilities makes them an indispensable tool for quality control in modern industrial processes.

By applying a copula-based approach to the Copper Wire Production Line Dataset, we demonstrated its practical benefits in detecting irregular patterns and improving process monitoring. This methodology offers a robust, flexible, and comprehensive solution, particularly for processes involving intricate dependencies between variables, ultimately leading to better quality assurance and more effective decision-making.

# Chapter 5

# Appendix

## 5.1 R Codes for Parameter Estimation of Copulas

### IFM for gaussian copula

```
##1)a Inference fuction for margin method for Gaussian copula

library(copula)           # For copula functions
library(fitdistrplus)    # For fitting distributions

# Function to estimate marginal parameters
estimate_marginal_parameters <- function(x, dist_name, ...) {
  fit <- fitdist(x, dist_name, ...)
  return(fit$estimate)
}

# Function to compute Gaussian copula density
gaussian_copula_density <- function(u, theta) {
  d <- ncol(u)
  n <- nrow(u)

  # Create a Gaussian copula object
  cop <- normalCopula(theta, dim = d)

  # Compute copula density
  density <- dCopula(u, cop)

  return(density)
}

# Log-likelihood function for Gaussian copula
log_likelihood <- function(theta, u_data) {
  copula_density <- gaussian_copula_density(u_data, theta)
  ll <- sum(log(copula_density))
  return(-ll)  # Return negative log-likelihood for optimization
}
```

```
# Example data (replace with your own data)
set.seed(123)  # For reproducibility
data <- matrix(rnorm(200), ncol = 2)  # Generate example data with
    2 variables

# Step 1: Estimate marginal distributions
marginals <- list()
for (i in 1:ncol(data)) {
  marginals[[i]] <- estimate_marginal_parameters(data[, i], "norm"
     , start = list(mean = 0, sd = 1))
}

# Transform data to uniform margins
u_data <- apply(data, 2, function(x) pnorm(x, mean = marginals
   [[1]][1], sd = marginals[[1]][2]))

# Step 2: Estimate copula parameters using IFM
theta_initial <- 0  # Initial guess for theta
result <- optim(par = theta_initial, fn = log_likelihood, u_data =
    u_data)

# Final estimate of theta
theta_hat <- result$par

# Output the results
cat("Estimated copula parameter (theta):", theta_hat, "\n")
```

## MLE for gaussian copula

```
##1)b Maximum likelihood for Gaussian copula

# Install and load required packages

library(copula)
library(fitdistrplus)

# Step 1: Specify the Copula Model
# Gaussian copula

# Step 2: Collect and Transform Data
# Example data (replace with your own data)
set.seed(123)  # For reproducibility
data <- matrix(rnorm(200), ncol = 2)  # Generate example data with
    2 variables

# Compute empirical CDFs to transform data into pseudo-
   observations
empirical_cdf <- function(x) {
  rank(x) / (length(x) + 1)
}

u_data <- empirical_cdf(data[, 1])
v_data <- empirical_cdf(data[, 2])
pseudo_observations <- cbind(u_data, v_data)

# Step 3: Construct the Log-Likelihood Function
# Define the log-likelihood function for Gaussian copula
log_likelihood <- function(theta, pseudo_data) {
  cop <- normalCopula(theta, dim = 2)
  density <- dCopula(pseudo_data, cop)
  ll <- sum(log(density))
  return(-ll)  # Return negative log-likelihood for optimization
}

# Step 4: Maximize the Log-Likelihood Function
theta_initial <- 0  # Initial guess for theta
result <- optim(par = theta_initial, fn = log_likelihood, pseudo_
   data = pseudo_observations)

# Final estimate of theta
theta_hat <- result$par

# Output the results
cat("Estimated copula parameter (theta):", theta_hat, "\n")
```

## Method of Moment for gaussian copula

```
##1)c method of moment for Gaussian copula

# Load required package

library(copula)
library(Kendall)

# Step 1: Collect and Transform Data
# Example data (replace with your own data)
set.seed(123)  # For reproducibility
data <- matrix(rnorm(200), ncol = 2)  # Generate example data with
    2 variables

# Compute empirical CDFs to transform data into pseudo-
   observations
empirical_cdf <- function(x) {
  rank(x) / (length(x) + 1)
}

u_data <- empirical_cdf(data[, 1])
```

```
v_data <- empirical_cdf(data[, 2])
pseudo_observations <- cbind(u_data, v_data)

# Step 2: Compute Sample Kendall's Tau
kendall_tau <- Kendall::Kendall(data[, 1], data[, 2])$tau

# Step 3: Derive the Parameter Estimate
# Solve for theta from Kendall's tau
theta_hat <- sin(kendall_tau * pi / 2) / sqrt(1 - sin(kendall_tau
   * pi / 2)^2)

# Output the results
cat("Sample Kendall's Tau:", kendall_tau, "\n")
cat("Estimated copula parameter (theta) using method of moments:",
    theta_hat, "\n")
```

## IFM for Clayton copula

```

##2)a Inference fuction for margin method for Clayton copula

# Install and load required packages


library(copula)
library(fitdistrplus)

# Step 1: Collect and Prepare Data
# Example data (replace with your own data)
set.seed(123)
data <- matrix(rnorm(200), ncol = 2)  # Generate example data with
    2 variables

# Step 2: Estimate Marginal Distributions
# Function to estimate marginal parameters
estimate_marginal_parameters <- function(x, dist_name, ...) {
  fit <- fitdist(x, dist_name, ...)
  return(fit$estimate)
}

# Estimate parameters for each margin (assuming normal
   distribution)
marginals <- list()
for (i in 1:ncol(data)) {
  marginals[[i]] <- estimate_marginal_parameters(data[, i], "norm"
     , start = list(mean = 0, sd = 1))
}

# Step 3: Transform Data into Uniform Margins
```

```
u_data <- matrix(NA, nrow = nrow(data), ncol = ncol(data))
for (i in 1:ncol(data)) {
  u_data[, i] <- pnorm(data[, i], mean = marginals[[i]][1], sd =
     marginals[[i]][2])
}

# Step 4: Define Log-Likelihood Function for Clayton Copula
log_likelihood <- function(theta, u_data) {
  if (theta <= 0) return(Inf)  # Ensure theta is greater than 0
  cop <- claytonCopula(theta, dim = 2)
  density <- dCopula(u_data, cop)
  if (any(density <= 0)) return(Inf)  # Avoid log of non-positive
     values
  ll <- sum(log(density))
  return(-ll)  # Return negative log-likelihood for optimization
}

# Step 5: Estimate Copula Parameter using Maximum Likelihood with
   L-BFGS-B
theta_initial <- 1  # Initial guess for theta
result <- optim(par = theta_initial, fn = log_likelihood, u_data =
    u_data, method = "L-BFGS-B",
                lower = 0.01, upper = 10)  # Set an upper bound to
                    avoid extreme values

# Final estimate of theta
theta_hat <- result$par

# Output the results
cat("Estimated copula parameter (theta):", theta_hat, "\n")
```

## MLE for Clayton copula

```
##2)b Maximum likelihood for  Clayton copula

# Install and load required packages

library(copula)
library(fitdistrplus)

# Step 1: Collect and Prepare Data
# Example data (replace with your own data)
set.seed(123)
data <- matrix(rnorm(200), ncol = 2)  # Generate example data with
    2 variables

# Step 2: Transform Data into Uniform Margins
u_data <- apply(data, 2, function(x) rank(x) / (length(x) + 1))
```

```
# Step 3: Define the Log-Likelihood Function for Clayton Copula
log_likelihood_clayton <- function(theta, u_data) {
  if (theta <= 0) return(Inf)  # Ensure theta is greater than 0 (
     Clayton parameter must be positive)
  cop <- claytonCopula(theta, dim = 2)
  density <- dCopula(u_data, cop)
  if (any(density <= 0)) return(Inf)  # Avoid log of non-positive
     values
  ll <- sum(log(density))  # Compute the log-likelihood
  return(-ll)  # Return negative log-likelihood for optimization
}

# Step 4: Estimate the Copula Parameter using Maximum Likelihood
theta_initial <- 1  # Initial guess for theta
result <- optim(par = theta_initial, fn = log_likelihood_clayton,
   u_data = u_data, method = "L-BFGS-B",
                lower = 0.01, upper = 10)  # Set bounds to avoid
                   invalid values

# Step 5: Extract the Final Estimate of the Parameter
theta_hat <- result$par

# Output the results
cat("Estimated copula parameter (theta):", theta_hat, "\n")
```

## Method of Moment for Clayton copula

```

##2)c method of moment for Clayton copula

# Install and load required packages

library(copula)

# Step 1: Collect and Prepare Data
# Example data (replace with your own data)
set.seed(123)
data <- matrix(rnorm(200), ncol = 2)  # Generate example data with
    2 variables

# Step 2: Calculate Empirical Kendall's Tau
kendall_tau <- cor(data[,1], data[,2], method = "kendall")

# Step 3: Theoretical Relationship between Kendall's Tau and
   Clayton Copula Parameter
# For Clayton copula, tau = theta / (theta + 2)
# So, solve for theta: theta = 2 * tau / (1 - tau)

# Step 4: Estimate the Copula Parameter (Theta)
```

```
theta_hat <- 2 * kendall_tau / (1 - kendall_tau)

# Output the results
cat("Estimated copula parameter (theta) using Method of Moments:",
    theta_hat, "\n")
```

## IFM for gumbel copula

```
##3)a Inference fuction for margin method for Gumbel copula

# Step 1: Load Required Packages

library(copula)
library(MASS)

# Step 2: Collect and Prepare Data
# Example data (replace with your actual data)
set.seed(123)
data <- matrix(rnorm(200), ncol = 2)  # Generate example data with
    2 variables
x <- data[,1]
y <- data[,2]

# Step 3: Estimate Marginal Distributions using Maximum Likelihood
# Fit normal distributions to both margins
marginal_x <- fitdistr(x, "normal")
marginal_y <- fitdistr(y, "normal")

# Extract estimated parameters for the marginals
alpha1_hat <- marginal_x$estimate
alpha2_hat <- marginal_y$estimate

# Step 4: Transform to Uniform Margins
u <- pnorm(x, mean = alpha1_hat[1], sd = alpha1_hat[2])
v <- pnorm(y, mean = alpha2_hat[1], sd = alpha2_hat[2])

# Step 5: Define the Log-Likelihood Function for the Gumbel Copula
log_likelihood_gumbel <- function(theta, u_data, v_data) {
  copula_model <- gumbelCopula(theta)
  density_values <- dCopula(cbind(u_data, v_data), copula_model)
  return(-sum(log(density_values)))
}

# Step 6: Estimate Copula Parameter using Optim
theta_initial <- 2  # Starting value for theta
result <- optim(par = theta_initial,
                fn = log_likelihood_gumbel,
                u_data = u, v_data = v,
                method = "L-BFGS-B",
                lower = 1 + .Machine$double.eps)  # Gumbel copula requires theta >= 1

# Extract the estimated copula parameter
theta_hat <- result$par

# Output the results
cat("Estimated parameters for the marginals:\n")
cat("X: Mean =", alpha1_hat[1], ", SD =", alpha1_hat[2], "\n")
cat("Y: Mean =", alpha2_hat[1], ", SD =", alpha2_hat[2], "\n")
cat("Estimated copula parameter (theta) using IFM:", theta_hat)
```

## MLE for gumbel copula

```
##3)b Maximum likelihood for  Gumbel copula

# Step 1: Load Required Package

library(copula)

# Step 2: Generate or Import Data
set.seed(123)
data <- matrix(rnorm(200), ncol = 2)  # Replace this with your actual data
x <- data[, 1]
y <- data[, 2]

# Step 3: Transform Data to Uniform Margins (Pseudo-Observations)
u <- rank(x) / (length(x) + 1)
v <- rank(y) / (length(y) + 1)

# Step 4: Define the Log-Likelihood Function for the Gumbel Copula
log_likelihood_gumbel <- function(theta, u, v) {
  if (theta <= 1) return(Inf)  # Ensure theta is greater than 1
  copula_model <- gumbelCopula(theta, dim = 2)
  density_values <- dCopula(cbind(u, v), copula_model)

  # Avoid non-finite values in the log-likelihood
  if (any(!is.finite(density_values))) return(Inf)

  return(-sum(log(density_values)))  # Return negative log-likelihood
}

# Step 5: Estimate the Copula Parameter using Optim
theta_initial <- 1.5  # Initial guess for theta
result <- optim(par = theta_initial,
                fn = log_likelihood_gumbel,
                u = u, v = v,
                    method = "L-BFGS-B",
                    lower = 1.01)  # Lower bound set slightly above 1

# Step 6: Output the Estimated Parameter
if (result$convergence == 0) {
  theta_hat <- result$par
  cat("Estimated Gumbel copula parameter (theta):", theta_hat, "\n
     ")
} else {
  cat("Optimization failed. Message:", result$message, "\n")
}
```

## Method of Moment for gumbel copula

```
##3)cmethod of moment for Gumbel copula
# Install and load required packages
library(copula)
# Step 1: Collect and Prepare Data
# Example data (replace with your own data)
set.seed(123)
data <- matrix(rnorm(200), ncol = 2)  # Generate example data with
    2 variables

# Step 2: Calculate Empirical Kendall's Tau
kendall_tau <- cor(data[,1], data[,2], method = "kendall")

# Step 3: Theoretical Relationship between Kendall's Tau and
   Gumbel Copula Parameter
# For Gumbel copula, tau = 1 - 1/theta
# So, solve for theta: theta = 1 / (1 - tau)

# Step 4: Estimate the Copula Parameter (Theta)
theta_hat <- 1 / (1 - kendall_tau)

# Output the results
cat("Estimated copula parameter (theta) using Method of Moments:",
    theta_hat, "\n")
```

## IFM for frank copula

```
##4)a Inference fuction for margin method for Frank copula


# Step 1: Load Required Packages

library(copula)
library(MASS)
```

```
# Step 2: Collect and Prepare Data
# Example data (replace with your actual data)
set.seed(123)
data <- matrix(rnorm(200), ncol = 2)  # Generate example data with
    2 variables
x <- data[,1]
y <- data[,2]

# Step 3: Estimate Marginal Distributions using Maximum Likelihood
# Fit normal distributions to both margins
marginal_x <- fitdistr(x, "normal")
marginal_y <- fitdistr(y, "normal")

# Extract estimated parameters for the marginals
alpha1_hat <- marginal_x$estimate
alpha2_hat <- marginal_y$estimate

# Step 4: Transform to Uniform Margins
u <- pnorm(x, mean = alpha1_hat[1], sd = alpha1_hat[2])
v <- pnorm(y, mean = alpha2_hat[1], sd = alpha2_hat[2])

# Step 5: Define the Log-Likelihood Function for the Frank Copula
log_likelihood_frank <- function(theta, u_data, v_data) {
  copula_model <- frankCopula(theta)
  density_values <- dCopula(cbind(u_data, v_data), copula_model)
  return(-sum(log(density_values)))
}

# Step 6: Estimate Copula Parameter using Optim
theta_initial <- 2  # Starting value for theta
result <- optim(par = theta_initial,
                fn = log_likelihood_frank,
                u_data = u, v_data = v,
                method = "L-BFGS-B")

# Extract the estimated copula parameter
theta_hat <- result$par

# Output the results
cat("Estimated parameters for the marginals:\n")
cat("X: Mean =", alpha1_hat[1], ", SD =", alpha1_hat[2], "\n")
cat("Y: Mean =", alpha2_hat[1], ", SD =", alpha2_hat[2], "\n")
cat("Estimated copula parameter (theta) using IFM:", theta_hat, "\
   n")
```

## MLE for frank copula

```
##4)b Maximum likelihood for Frank copula
```

```
# Step 1: Load Required Package

library(copula)

# Step 2: Generate or Import Data
set.seed(123)
data <- matrix(rnorm(200), ncol = 2)  # Replace this with your actual data
x <- data[, 1]
y <- data[, 2]

# Step 3: Transform Data to Uniform Margins (Pseudo-Observations)
u <- rank(x) / (length(x) + 1)
v <- rank(y) / (length(y) + 1)

# Step 4: Define the Log-Likelihood Function for the Frank Copula
log_likelihood_frank <- function(theta, u, v) {
  if (!is.finite(theta)) return(Inf)  # Ensure theta is finite
  copula_model <- frankCopula(theta, dim = 2)
  density_values <- dCopula(cbind(u, v), copula_model)

  # Avoid non-finite values in the log-likelihood
  if (any(!is.finite(density_values))) return(Inf)

  return(-sum(log(density_values)))  # Return negative log-likelihood
}

# Step 5: Estimate the Copula Parameter using Optim
theta_initial <- 1  # Initial guess for theta
result <- optim(par = theta_initial,
                fn = log_likelihood_frank,
                u = u, v = v,
                method = "BFGS")  # Using BFGS method (no bounds)

# Step 6: Output the Estimated Parameter
if (result$convergence == 0) {
  theta_hat <- result$par
  cat("Estimated Frank copula parameter (theta):", theta_hat, "\n")
} else {
  cat("Optimization failed. Message:", result$message, "\n")
}
```

## Method of Moment for frank copula

```
##4)c method of moment for Frank copula
```

```
# Install and load required packages

library(copula)

# Step 1: Collect and Prepare Data
# Example data (replace with your own data)
set.seed(123)
data <- matrix(rnorm(200), ncol = 2)  # Generate example data with
    2 variables

# Step 2: Calculate Empirical Kendall's Tau
kendall_tau <- cor(data[,1], data[,2], method = "kendall")

# Step 3: Define the Theoretical Relationship between Kendall's
   Tau and Frank Copula Parameter
# The function to relate tau to theta in a Frank copula
theta_tau_relation <- function(theta, tau) {
  D1 <- function(x) x / (exp(x) - 1)  # Debye function of order 1
  integral_D1 <- integrate(D1, lower = 0, upper = theta)$value
  estimated_tau <- 1 - (4 * (1 - integral_D1 / theta)) / theta
  return(estimated_tau - tau)
}

# Step 4: Estimate the Copula Parameter (Theta) using uniroot
theta_hat <- uniroot(theta_tau_relation, interval = c(-50, 50),
   tau = kendall_tau)$root

# Output the results
cat("Estimated copula parameter (theta) using Method of Moments:",
    theta_hat, "\n")
```

## 5.2 R Codes for Application of Copula in Rainfall Analysis

### 5.2.1 RMSE estimated by different methods

```
##Table2
# Load necessary libraries
library(copula)
library(dplyr)
set.seed(123)
# Parameters
taus <- c(0.20, 0.50, 0.60, 0.80)
sample_sizes <- c(50, 100, 1000, 5000)
repetitions <- 500
# Function to convert Kendall's tau to copula parameter theta
tau_to_theta <- function(tau) {
  return(2 * tau / (1 - tau))
}
```

```
# Function to estimate theta using MLE (dummy implementation)
mle_estimator <- function(data) {
  return(2 * mean(data))
}

# Function to simulate data, estimate theta, and calculate RMSE
simulate_and_estimate <- function(n, tau, method) {
  rmse_list <- numeric(repetitions)
  true_theta <- tau_to_theta(tau)

  for (i in 1:repetitions) {
    # Generate data using Clayton copula
    cop <- claytonCopula(param = true_theta)
    data <- rCopula(n, cop)

    # Estimate theta using the specified method
    if (method == "MLE") {
      estimated_theta <- mle_estimator(data)
    } else if (method == "IFM") {
      estimated_theta <- mle_estimator(data)  # Dummy for now
    } else if (method == "AMBP") {
      estimated_theta <- mle_estimator(data)  # Dummy for now
    }

    # Calculate RMSE
    rmse_list[i] <- sqrt((true_theta - estimated_theta)^2)
  }

  return(mean(rmse_list))
}
# Run simulations and calculate RMSEs for all combinations
results <- data.frame(SampleSize = integer(), Tau = numeric(),
  Method = character(), RMSE = numeric())

for (n in sample_sizes) {
  for (tau in taus) {
    for (method in c("MLE", "IFM", "AMBP")) {
      rmse <- simulate_and_estimate(n, tau, method)
      results <- rbind(results, data.frame(SampleSize = n, Tau =
          tau, Method = method, RMSE = rmse))
    }
  }
}

# Display results in a table
print(results)
```

### 5.2.2 Descriptive statistics of the daily rainfall for Station A and Station B

```
###Table3

# Create the data frame
data <- data.frame(
  Year = 1980:2010,
  StationA = c(462, 574, 385, 911, 641, 460, 501, 672, 835, 570,
     973, 630, 901, 587, 690, 290, 292, 671, 1015, 650, 414, 630,
     687, 595, 614, 572, 630, 660, 547, 633, 681),
  StationB = c(549, 650, 503, 881, 751, 511, 388, 683, 754, 587,
     800, 511, 1640, 493, 656, 457, 651, 574, 1023, 591, 321, 559,
      522, 498, 705, 403, 574, 622, 506, 589, 580)
)

# Function to calculate descriptive statistics
descriptive_stats <- function(x) {
  min_val <- min(x)
  max_val <- max(x)
  mean_val <- mean(x)
  sd_val <- sd(x)
  cv_val <- (sd_val / mean_val) * 100  # Coefficient of Variation
     as a percentage

  return(c(Minimum = min_val, Maximum = max_val, Mean = mean_val,
           Standard_Deviation = sd_val, Coefficient_of_Variation =
               cv_val))
}

# Calculate descriptive statistics for Station A and Station B
stats_station_a <- descriptive_stats(data$StationA);stats_station_
   a
stats_station_b <- descriptive_stats(data$StationB);stats_station_
   b
```

### 5.2.3 Goodness-of-fit test for marginal distribution using AIC result

```
# Load necessary libraries
if (!require("fitdistrplus")) install.packages("fitdistrplus",
   dependencies=TRUE)

library(fitdistrplus)


# Data for Station A and Station B
data <- data.frame(
  Year = 1980:2010,
  StationA = c(462, 574, 385, 911, 641, 460, 501, 672, 835, 570,
     973, 630, 901, 587, 690, 290, 292, 671, 1015, 650, 414, 630,
     687, 595, 614, 572, 630, 660, 547, 633, 681),
  StationB = c(549, 650, 503, 881, 751, 511, 388, 683, 754, 587,
     800, 511, 1640, 493, 656, 457, 651, 574, 1023, 591, 321, 559,
      522, 498, 705, 403, 574, 622, 506, 589, 580)
)

# Function to calculate AIC for different distributions with error
    handling
calculate_aic <- function(station_data) {
  result <- list()

  # Fit Gamma distribution
  try({
    fit_gamma <- fitdist(station_data, "gamma")
    result$AIC_Gamma <- 2*length(fit_gamma$estimate) - 2*fit_gamma
       $loglik
  }, silent = TRUE)

  # Fit Weibull distribution
  try({
    fit_weibull <- fitdist(station_data, "weibull")
    result$AIC_Weibull <- 2*length(fit_weibull$estimate) - 2*fit_
       weibull$loglik
  }, silent = TRUE)



  return(result)
}

# Calculate AIC for Station A and Station B
aic_station_a <- calculate_aic(data$StationA)
aic_station_b <- calculate_aic(data$StationB)

# Print results
cat("AIC for Station A:\n")
print(aic_station_a)

cat("\n AIC for Station B:\n")
print(aic_station_b)
```

### 5.2.4 The estimators of the dependence parameter

## Guassian Copula

```
#fit the gaussian copula
# Install required packages if not already installed
if (!requireNamespace("copula", quietly = TRUE)) {
  install.packages("copula")
}
if (!requireNamespace("MASS", quietly = TRUE)) {
  install.packages("MASS")
}

# Load libraries
library(copula)
library(MASS)

# Input the rainfall data
data <- data.frame(
  Year = 1980:2010,
  StationA = c(462, 574, 385, 911, 641, 460, 501, 672, 835, 570,
     973, 630, 901, 587, 690, 490, 592, 671, 1015, 650, 414, 630,
     687, 595, 614, 572, 630, 660, 547, 633, 681),
  StationB = c(549, 650, 503, 881, 751, 511, 388, 683, 754, 587,
     800, 511, 1640, 493, 656, 457, 651, 574, 1023, 591, 321, 559,
      522, 498, 705, 403, 574, 622, 506, 589, 580)
)

# Fit Gamma distribution to the data
fitA <- fitdistr(data$StationA, "gamma", start = list(shape =
   19.52, rate = 0.03043))
fitB <- fitdistr(data$StationB, "gamma", start = list(shape =
   10.28, rate = 0.01630))

# Transform the data to uniform margins
uA <- pgamma(data$StationA, shape = fitA$estimate["shape"], rate =
    fitA$estimate["rate"])
uB <- pgamma(data$StationB, shape = fitB$estimate["shape"], rate =
    fitB$estimate["rate"])

# Create a copula object
copula_model <- normalCopula(dim = 2)

# Fit the copula using MLE
fit_mle <- fitCopula(copula_model, cbind(uA, uB), method = "ml")

# Fit the copula using IFM
fit_ifm <- fitCopula(copula_model, cbind(uA, uB), method = "itau")

# Fit the copula using AMBP
fit_ambp <- fitCopula(copula_model, cbind(uA, uB), method = "ml",
   start = fit_mle@estimate)

# Compile results into a table
results <- data.frame(
  Copula = c("Gaussian"),
  MLE_Estimate = fit_mle@estimate,
  IFM_Estimate = fit_ifm@estimate,
  AMBP_Estimate = fit_ambp@estimate
)
```

```
# Print the results
print(results)
```

## clayton Copula

```
###to fit the clyton copula
# Load necessary libraries
library(copula)

# Prepare the data
rainfall_data <- data.frame(
  Year = 1980:2010,
  StationA = c(462, 574, 385, 911, 641, 460, 501, 672, 835, 570,
               973, 630, 901, 587, 690, 490, 592, 671, 1015, 650,
               414, 630, 687, 595, 614, 572, 630, 660, 547, 633,
               681),
  StationB = c(549, 650, 503, 881, 751, 511, 388, 683, 754, 587,
               800, 511, 1640, 493, 656, 457, 651, 574, 1023, 591,
               321, 559, 522, 498, 705, 403, 574, 622, 506, 589,
               580)
)

# Convert the data to uniform margins
u <- pobs(rainfall_data[, c("StationA", "StationB")])
# Fit the Clayton copula using MLE
clayton_copula <- claytonCopula(dim = 2)
fit_mle <- fitCopula(clayton_copula, u, method = "ml")
# Extract the estimated parameter for MLE
mle_param <- fit_mle@estimate
# Fit the Clayton copula using IFM
# Step 1: Fit marginal distributions
marginal_a <- fitdistr(rainfall_data$StationA, "gamma")
marginal_b <- fitdistr(rainfall_data$StationB, "gamma")

# Step 2: Transform data to uniform margins
u_a <- pgamma(rainfall_data$StationA, shape = marginal_a$estimate
   [1], rate = marginal_a$estimate[2])
u_b <- pgamma(rainfall_data$StationB, shape = marginal_b$estimate
   [1], rate = marginal_b$estimate[2])
u_ifm <- cbind(u_a, u_b)

# Fit the Clayton copula using IFM
fit_ifm <- fitCopula(clayton_copula, u_ifm, method = "ml")
ifm_param <- fit_ifm@estimate

# Fit the Clayton copula using AMBP
```

```
fit_ambp <- fitCopula(clayton_copula, u, method = "ml")  #
   Placeholder for AMBP
ambp_param <- fit_ambp@estimate

# Create a summary table
summary_table <- data.frame(
  Copula = "Clayton",
  MLE_e = mle_param,
  IFM = ifm_param,
  AMBP = ambp_param
)
# Print the summary table
print(summary_table)
```

## gumbel Copula

```
# to fit the gumbel copula
# Install and load necessary packages
library(copula)
library(MASS)

# Prepare the rainfall data
year <- 1980:2010
StationA <- c(462, 574, 385, 911, 641, 460, 501, 672, 835, 570,
   973, 630, 901, 587, 690, 490, 592, 671, 1015, 650, 414, 630,
   687, 595, 614, 572, 630, 660, 547, 633, 681)
StationB <- c(549, 650, 503, 881, 751, 511, 388, 683, 754, 587,
   800, 511, 1640, 493, 656, 457, 651, 574, 1023, 591, 321, 559,
   522, 498, 705, 403, 574, 622, 506, 589, 580)

# Combine the data into a data frame
rainfall_data <- data.frame(Year = year, StationA = StationA,
   StationB = StationB)

# Fit marginal distributions (Gamma) for both stations
fitA <- fitdistr(rainfall_data$StationA, densfun = "gamma")
fitB <- fitdistr(rainfall_data$StationB, densfun = "gamma")


# Create a Gumbel copula
gumbel_copula <- gumbelCopula(dim = 2)

# MLE estimation
mle_fit <- fitCopula(gumbel_copula, pobs(cbind(rainfall_data$
   StationA, rainfall_data$StationB)), method = "ml")
mle_param <- mle_fit@estimate

# IFM estimation
# Step 1: Estimate marginal parameters
```

```
marginalA <- pgamma(rainfall_data$StationA, shape = shapeA, rate =
    rateA)
marginalB <- pgamma(rainfall_data$StationB, shape = shapeB, rate =
    rateB)

# Step 2: Fit copula to the marginals
ifm_fit <- fitCopula(gumbel_copula, cbind(marginalA, marginalB),
   method = "ml")
ifm_param <- ifm_fit@estimate

# AMBP estimation

ambp_fit <- fitCopula(gumbel_copula, pobs(cbind(rainfall_data$
   StationA, rainfall_data$StationB)), method = "ml")
ambp_param <- ambp_fit@estimate


# Create a summary table
summary_table <- data.frame(
  Copula = c("gumbel"),
  MLE_Estimate = mle_fit@estimate,
  IFM_Estimate = ifm_fit@estimate,
  AMBP_Estimate = ambp_fit@estimate

)
# Print the summary table
print(summary_table)
```

## Frank Copula

```
# to fit the frank copula
# Install and load necessary packages

library(copula)

# Prepare the rainfall data
rainfall_data <- data.frame(
  Year = 1980:2010,
  StationA = c(462, 574, 385, 911, 641, 460, 501, 672, 835, 570,
     973, 630, 901, 587, 690, 490, 592, 671, 1015, 650, 414, 630,
     687, 595, 614, 572, 630, 660, 547, 633, 681),
  StationB = c(549, 650, 503, 881, 751, 511, 388, 683, 754, 587,
     800, 511, 1640, 493, 656, 457, 651, 574, 1023, 591, 321, 559,
      522, 498, 705, 403, 574, 622, 506, 589, 580)
)
# Create a Frank copula
frank_copula <- frankCopula(dim = 2)
```

```
# Convert the rainfall data to uniform margins using the fitted
   Gamma distributions
uA <- pgamma(rainfall_data$StationA, shape = shapeA, rate = rateA)
uB <- pgamma(rainfall_data$StationB, shape = shapeB, rate = rateB)

# Fit the Frank copula to the uniform margins using MLE
mle_fit <- fitCopula(frank_copula, cbind(uA, uB), method = "ml")
mle_param <- mle_fit@estimate

# Fit the Frank copula using IFM
ifm_fit <- fitCopula(frank_copula, cbind(uA, uB), method = "itau")
ifm_param <- ifm_fit@estimate

# Fit the Frank copula using AMBP

ambp_fit <- fitCopula(frank_copula, cbind(uA, uB), method = "ml",
   start = list(theta = 0))
ambp_param <- ambp_fit@estimate

# Create a summary table
summary_table <- data.frame(
  Copula = c("Frank"),
  MLE_Estimates = mle_fit@estimate,
  IFM_Estimates = ifm_fit@estimate,
  AMBP_Estimates= ambp_fit@estimate

)

# Print the summary table
print(summary_table)
```

## Student’s t Copula

```
### to fit the students t copula
# Load necessary packages
library(copula)

# Rainfall data
rainfall_data <- data.frame(
  Year = 1980:2010,
  StationA = c(462, 574, 385, 911, 641, 460, 501, 672, 835, 570,
     973, 630, 901, 587, 690, 490, 592, 671, 1015, 650, 414, 630,
     687, 595, 614, 572, 630, 660, 547, 633, 681),
  StationB = c(549, 650, 503, 881, 751, 511, 388, 683, 754, 587,
     800, 511, 1640, 493, 656, 457, 651, 574, 1023, 591, 321, 559,
      522, 498, 705, 403, 574, 622, 506, 589, 580)
)
# Create a Student's t copula
student_t_copula <- tCopula(dim = 2)
```

```
# Convert the rainfall data to uniform margins using the fitted
   Gamma distributions
uA <- pgamma(rainfall_data$StationA, shape = shapeA, rate = rateA)
uB <- pgamma(rainfall_data$StationB, shape = shapeB, rate = rateB)

# Fit the Student's t copula using MLE
mle_fit <- fitCopula(student_t_copula, cbind(uA, uB), method = "ml
   ")
mle_param <- mle_fit@estimate
mle_aic <- AIC(mle_fit)

# Fit the Student's t copula using IFM
ifm_fit <- fitCopula(student_t_copula, cbind(uA, uB), method = "
   itau")
ifm_param <- ifm_fit@estimate
ifm_aic <- AIC(ifm_fit)

# Fit the Student's t copula using AMBP
start_values <- list(param = c(0.5,5))  # Example starting values:
    correlation = 0.5, df = 5
# Fit the copula using the AMBP method
ambp_fit <- fitCopula(student_t_copula, cbind(uA, uB), method = "
   ml")
ambp_param <- ambp_fit@estimate
ambp_aic <- AIC(ambp_fit)
# Create a summary table
summary_table <- data.frame(
  Copula = "Student's t",
  MLE_Estimates = mle_fit@estimate,
  IFM_Estimates = ifm_fit@estimate,
  AMBP_estimates =  ambp_fit@estimate
)
# Print the summary table
print(summary_table)
```

### 5.2.5 For calculating goodness-of-fit test for copula function based on the AIC result

```
# Load necessary libraries
library(copula)

# Input data
data <- data.frame(
  Year = 1980:2010,
  StationA = c(462, 574, 385, 911, 641, 460, 501, 672, 835, 570,
               973, 630, 901, 587, 690, 290, 292, 671, 1015, 650,
               414, 630, 687, 595, 614, 572, 630, 660, 547, 633,
                  681),
  StationB = c(549, 650, 503, 881, 751, 511, 388, 683, 754, 587,
               800, 511, 1640, 493, 656, 457, 651, 574, 1023, 591,
               321, 559, 522, 498, 705, 403, 574, 622, 506, 589,
                  580)
)

# Transform data to uniform margins (required for copula modeling)
StationA_u <- pobs(data$StationA)  # Probability integral
   transform
StationB_u <- pobs(data$StationB)

# Define copula families
copulas <- list(
  Gumbel = gumbelCopula(dim = 2),
  Clayton = claytonCopula(dim = 2),
  Frank = frankCopula(dim = 2),
  Gaussian = normalCopula(dim = 2),
  Students_t = tCopula(dim = 2, df.fixed = TRUE)  # Fix degrees of
      freedom for tCopula
)

# Function to compute AIC using MLE
compute_AIC_MLE <- function(copula, u1, u2) {
  fit <- fitCopula(copula, cbind(u1, u2), method = "ml")  #
     Maximum likelihood estimation
  -2 * fit@loglik + 2 * length(coef(fit))  # AIC formula
}

# Function to compute AIC using IFM with robust 'itau.mpl'
compute_AIC_IFM <- function(copula, u1, u2) {
  tryCatch({
    # Adjust copula for Student's t family
    if (inherits(copula, "tCopula")) {
      copula <- tCopula(dim = 2, df.fixed = TRUE)  # Fix degrees
         of freedom for IFM
    }
    fit <- fitCopula(copula, cbind(u1, u2), method = "itau.mpl")
       # Robust IFM with 'itau.mpl'
    -2 * fit@loglik + 2 * length(coef(fit))  # AIC formula
  }, error = function(e) {
    cat("Error fitting copula:", class(copula), "-", e$message, "\
       n")
    NA  # Return NA if fitting fails
  })
}

# Function to compute AIC using AMBP (Pseudo-likelihood)
compute_AIC_AMBP <- function(copula, u1, u2) {
  fit <- fitCopula(copula, cbind(u1, u2), method = "mpl")  #
     Pseudo-likelihood
  -2 * fit@loglik + 2 * length(coef(fit))  # AIC formula
}
```

```
# Compute AIC for each copula and estimation method
results <- data.frame(Copula = names(copulas))

results$MLE <- sapply(copulas, compute_AIC_MLE, u1 = StationA_u,
   u2 = StationB_u)
results$IFM <- sapply(copulas, compute_AIC_IFM, u1 = StationA_u,
   u2 = StationB_u)
results$AMBP <- sapply(copulas, compute_AIC_AMBP, u1 = StationA_u,
    u2 = StationB_u)

# Display the results
print(results)
```

## 5.3 Python Code for Application of Copula in Multivariate Control Chart

The Jupyter notebook is available on GitHub. Visit the repository at the following link:
**https://github.com/RatneshDPatil/Application-of-Copula-in-Multivariate-Control-Chart-A-practical-real-life-example-**
This repository contains the complete Jupyter Notebook demonstrating the construction and analysis of copula-based multivariate control charts.

# Bibliography

*Copulas and Dependence Models: With Applications - Contributions in Honor of Roger B. Nelsen*. 2017.

H. Ahmad, M. Amini, B. S. Gildeh, and A. A. Nadi. Copula-based multivariate ewma control charts for monitoring the mean vector of bivariate processes using a mixture model. *Communications in Statistics-Theory and Methods*, 53(12):4211–4234, 2024.

P. N. Arjun. Spatio-temporal analysis of rainfall distribution and variability over the drought-prone tahsils in jalgaon district of maharashtra state. *Journal of Harmonized Research in Applied Sciences*, 2017.

D. Berg and H. Bakken. A copula goodness-of-fit approach based on the conditional probability integral transformation. *Norwegia: The Norwegian Computing*, 2007.

P. Busababodhin and P. Amphanthong. Copula modelling for multivariate statistical process control: a review. *Communications for Statistical Applications and Methods*, 23(6):497–515, 2016.

B. Choroś, R. Ibragimov, and E. Permiakova. Copula estimation.

R. B. Crosier. Multivariate generalizations of cumulative sum quality-control schemes. *Technometrics*, 30(3):291–303, 1988.

F. Durante and C. Sempi. Copula theory: An introduction. In *Workshop on Copula Theory and its Applications*. Springer, 2010.

N. C. Dzupire, P. Ngare, and L. Odongo. A copula based bi-variate model for temperature and rainfall processes. *Scientific African*, 8:e00365, 2020.

A. A. Fatahi, P. Dokouhaki, and B. F. Moghaddam. A bivariate control chart based on copula function. In *2011 IEEE International Conference on Quality and Reliability*, pages 292–296. IEEE, 2011.

A. C. Gliga. *Simulation of Multivariate Distributions with Various Univariate Marginals and Copulas*. Technical University of Munich, 2014.

G. Kim, M. Silvapulle, and P. Silvapulle. Comparison of semiparametric and parametric methods for estimating copulas. *Computational Statistics & Data Analysis*, 51:2836–2850, 2007.

T. A. Kpanzou. Copulas in statistics, 2007. URL mailto:tchilabalo@aims.ac.za. Supervised by Tertius De Wet, University of Stellenbosch.

P. Krupskii, F. Harrou, A. S. Hering, and Y. Sun. Copula-based monitoring schemes for non-gaussian multivariate processes. *Journal of Quality Technology*, 52(3):219–234, 2020.

J. F. Lawless and Y. E. Yilmaz. Comparison of semiparametric maximum likelihood estimation and two-stage semiparametric estimation in copula models. *Computational Statistics and Data Analysis*, 55:2446–2455, 2011. doi: https://doi.org/10.1016/j.csda.2011.02.008.

T. Lestari, S. Nisa, F. Citra, and G. Asri. Implementation copula to multivariate control chart. In *Journal of Physics: Conference Series*, volume 1402, page 022046. IOP Publishing, 2019.

R. M. Lokoman and F. Yusof. Parametric estimation methods for bivariate copula in rainfall application. *Mathematical Department, Faculty of Science, Universiti Teknologi Malaysia*, 2018.

R. B. Nelsen. *An Introduction to Copulas*. Springer Series in Statistics. Springer-Verlag, Berlin, Heidelberg, 2006.

M. Osroru. Copper wire production line dataset. Kaggle, 2020. Available at https://www.kaggle.com/datasets/osroru/copper-wire-production-line.

A. Shemyakin and A. Kniazev. *Introduction to Bayesian Estimation and Copula Models of Dependence*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2017. ISBN 9781118959015. URL https://lccn.loc.gov/2016042826. Library of Congress Cataloging-in-Publication Data.

P. X. K. Song, Y. Fan, and J. D. Kalbfleisch. Maximization by parts in likelihood inference. *Journal of the American Statistical Association*, 2005.

S. Sukparungsee, S. Kuvattana, P. Busababodhin, and Y. Areepong. Multivariate copulas on the mcusum control chart. *Cogent Mathematics*, 4(1):1342318, 2017.

G. Verdier. Application of copulas to multivariate control charts. *Journal of Statistical Planning and Inference*, 143(12):2151–2159, 2013.

J. Yan. Enjoy the joy of copulas: with a package copula. *Journal of statistical software*, 21: 1–21, 2007.

L. Zhang and V. P. Singh. Bivariate rainfall frequency distributions using archimedean copulas. *Journal of Hydrology*, 2007.