Examining Rice Grain Characteristics Using a Semi-Parametric Gaussian Model Approach

Matthew Coe
The Australian National University

February 19, 2024

This project was awarded a score of 38/40 in the course STAT3016.

Abstract

A Semi-parametric Gaussian Copula model is used to analyse rice characteristics. MCMC samples are generated with a scaled inverse-Wishart prior distribution combined with an extended rank likelihood. The findings indicate that these dependencies are interconnected and influence the classification of rice varieties.

1 Introduction

"How do the features of rice grains interdepend, and can understanding these dependencies provide insights into the distinct characteristics of different rice varieties?"

Rice ranks among the top grain products globally, following wheat and corn. It plays a vital role in human nutrition and has economic significance. Quality criteria for rice include factors like physical appearance, cooking characteristics, and taste. In this paper, we employ the Gaussian copula model to analyse these feature interdependencies. By mapping out the joint distribution of the rice grain features, we can identify patterns that signify the relationships between various features.

The dataset used in this research is the "Rice MSC Dataset [1]." "Determination of Effective and Specific Physical Features of Rice Varieties by Computer Vision In Exterior Quality Inspection," conducted by İlkay Çinar and Murat Koklu in 2021, image processing techniques were utilized to extract features from five different rice varieties of the same brand [2]. The dataset comprises data from 74,992 rice grains, with around 15,000 grains representing each variety. After preprocessing the images, 106 features were derived, including 12 morphological features, 4 shape features using the morphological data, and 90 color features from five distinct color spaces (RGB, HSV, Lab*, YCbCr, XYZ).

1.1 Model Selection

Due to not being computationally feasible, we must reduce the variable size from 106 to 6, not including our classification variable `CLASS`. To do this we will select variables on a prior belief that these will have the most influence on classification. The variables can be separated into 3 categories; Morphological: 12, Shape: 6, and Colour: 90.

Morphological:

`MAJORAXIS`: The length of the major axis can be a distinguishing factor in identifying long-grained vs. short-grained rice varieties. Different rice varieties can differ significantly in grain length.

`ROUNDNESS`: The roundness can provide insights into the overall shape of the rice grain. Some rice varieties might be more rounded while others might be slender.

Shape:

`SHAPEFACTOR1`: It is calculated by dividing the major axis length by the area.
This factor directly incorporates the major axis length, which is an essential attribute to capture the elongation of rice grains, a distinguishing feature among different rice varieties.

`SHAPEFACTOR4`: The equation is:

$$SF_4 = \frac{A}{(\frac{L}{2})^2\pi} \tag{1}$$

Given that this feature incorporates both the major and minor axis lengths squared, it provides a comprehensive understanding of the grain's shape by comparing it to a full ellipse defined by both axes. This could be useful in differentiating varieties that might have similar lengths but differ in overall shape.

Colour:

`meanL`: L* in the Lab* color space represents the lightness. It provides a measure that is closer to human perception of brightness. Analyzing the mean lightness across different rice grain images might provide valuable insights into the color characteristics of the grains.

`entropyH`: The entropy of the Hue component in the HSV color space can capture the variations in color shades and intensities. Given that Hue represents the type of color (like red, blue, or yellow), the entropy in this channel can be particularly useful in identifying slight variations in the color of rice grains, which might be indicative of different varieties.

Our rationale for these variables was that they should capture the most distinctive and meaningful characteristics of the data, providing insights that differentiate between different

classes effectively.

2 Gaussian Copula Model

To distinguish between various rice varieties accurately, it's important understand the underlying relationships between the rice grain features. A Gaussian Copula model, by emphasizing dependencies between these features, presents an effective approach to address this requirement.

One of the significant advantages of the Gaussian Copula model is its ability to model dependencies without imposing stringent assumptions on the univariate marginal distributions of each feature, enabling us to focus primarily on the multivariate relationships.

2.1 Model

In this study, we utilize a Semiparametric Gaussian Copula model to investigate the dependencies between features of rice grains.

The choice of a semi-parametric model is to balance the structure of parametric methods with the flexibility of non-parametric ones, making them ideal for high-dimensional datasets like rice grain features.

The model is characterized by its associated correlation matrix C.

$$z_1, \ldots, z_n \mid C \sim \text{i.i.d. multivariate normal}(0, C) \tag{2}$$

$$y_{i,j} = F_j^{-1}[\Phi(z_{i,j})] = G_j(z_{i,j}) \tag{3}$$

Here, F_j^{-1} or equivalently G_j denotes the (pseudo) inverse of an unknown univariate CDF, which may not necessarily be continuous. The $y_{i,j}$ values represent the transformed data points of $z_{i,j}$ after applying the Gaussian copula transformation. This approach ensures that $y_{i,j}$ maintains the dependency structure encapsulated by the Gaussian copula, irrespective of the marginal distributions of its components [3].

The Gaussian copula can be further expressed as:

$$Gauss_C(u) := \Phi_C(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_d)) \tag{4}$$

The significance of this correlation matrix is that it captures the pairwise dependencies between the features, which is crucial in understanding the relationships among rice grain features.

2.1.1 Priors

The prior distribution for C is defined based on a latent variable V as:

$$V \sim \text{inverse-Wishart}(\nu_0, \nu_0 V_0) \tag{5}$$

where it is parameterized such that $E[V^{-1}] = V_0^{-1}$.

The correlation matrix C is then defined to be equal in distribution to the matrix with entries given by:

$$C[i,j] = \frac{V[i,j]}{\sqrt{V[i,i]V[j,j]}} \tag{6}$$

2.2 Method

For Bayesian inference of the correlation matrix C, we implement a Markov Chain Monte Carlo (MCMC) technique, specifically the Gibbs sampling algorithm. By constructing a Markov chain with a stationary distribution proportional to $p(C) \times p(Z \in D|C)$, we can obtain samples for C and make posterior inferences [3].

Hoff (2007) provides an outline for a Gibbs sampling as such:

1. Resample Z: Iteratively over (i,j), sample $z_{i,j}$ from $p(z_{i,j}|V, Z[-i,-j], Z \in D)$ as follows:

For each $j \in \{1,\ldots,p\}$ and for each y in unique$\{y_{1,j},\ldots,y_{n,j}\}$:

1. Compute $z_l = \max\{z_{i,j} : y_{i,j} < y\}$ and $z_u = \min\{z_{i,j} : y < y_{i,j}\}$.
2. For each i such that $y_{i,j} = y$:
   (a) Compute $\sigma_j^2 = V[j,j] - V[j,-j]V^{-1}[-j,-j]V[-j,j]$.
   (b) Compute $\mu_{i,j} = Z[i,-j](V[j,-j]V^{-1}[-j,-j])^T$.
   (c) Sample $u_{i,j}$ uniformly from $(\Phi\left[\frac{z_l-\mu_{i,j}}{\sigma_j}\right], \Phi\left[\frac{z_u-\mu_{i,j}}{\sigma_j}\right])$.
   (d) Set $z_{i,j} = \mu_{i,j} + \sigma_j \times \Phi^{-1}(u_{i,j})$.

2. Resample V: Sample V from an inverse-Wishart$(\nu_0 + n, \nu_0 V_0 + Z^T Z)$ distribution.

3. Compute C: Let $C[i,j] = \frac{V[i,j]}{\sqrt{V[i,i]V[j,j]}}$.

Iteration of this algorithm generates a Markov chain in C whose stationary distribution is $p(C|Z \in D)$. While not apparent in our dataset, this algorithm is easily modified to accommodate data that are missing-at-random: If $y_{i,j}$ is missing, the full conditional distribution of $z_{i,j}$ is the unconstrained normal distribution with mean $\mu_{i,j}$ and variance σ_j^2 given above.

2.2.1 sbgcop

'sbgcop' is an R package designed for simulating and analyzing multivariate data via copula methods. Developed by Hoff [4], this package provides tools for estimating and inferring parameters within a Gaussian copula model. Although the package can be sourced from CRAN, it was installed using devtools directly from the author's GitHub repository.

The main function of sbgcop is 'sbgcop.mcmc', which generates MCMC samples from the posterior distribution of a correlation matrix C. This function employs a scaled inverse-Wishart prior distribution in conjunction with an extended rank likelihood.

A notable feature of the package is its capability to regard univariate marginal distributions as nuisance parameters, streamlining the process for specific research needs. Even though our analysis did not require it, it's worth noting that sbgcop comes equipped with a semiparametric imputation procedure, proving beneficial for projects dealing with missing data in multivariate datasets.

The analyses were conducted on a Ubuntu 22.04.2 LTS operating system with x86_64 architecture, using R version 4.1.2. The model was fitted using the 'sbgcop.mcmc' function, running a total of 75,000 iterations and taking 68 minutes and 20 seconds to run. The number of saved samples is 1000. The main output produced was 'C.psamp'. This refers to the collection of the correlation matrix samples. 'C.psamp' is structured as an array of size p x p x nsamp, representing the posterior samples of the correlation matrix. Each slice of this 3-dimensional array gives a sampled correlation matrix from the posterior distribution.

2.3 Simulation

We apply this model to a simulation designed to derive a predictive sample of y, mirroring the inter-dependencies and structures inherent in the observed data.

Our sampling procedure is:

1. Draw a sample from the posterior distribution of C, given by:

$$C \sim p(C|Z \in D) \tag{7}$$

2. Obtain a sample z from a multivariate normal distribution with mean 0 and covariance matrix C (1).

3. Utilize the empirical univariate marginal distributions $\hat{F}_j$ of the observed data to transform the samples (2) [1].

[1]Specifically, for each simulated multivariate sample, the transformation is done using the quantile function in R, which leverages the empirical CDFs of each feature in the dataset. This ensures that the simulated data maintains the same univariate marginal distributions as the observed data.

This method combines the information from the posterior distribution of C with the empirical marginal distributions of the observed data to produce a unified representation. Notably, while this approach offers a predictive joint distribution aligned with the observed data's univariate marginal distributions, it avoids uncertainties inherent in estimating $F_1, ..., F_p$ (marginal CDFs) for predicting y but not C. Consequently, the multivariate dependence is depicted through the Gaussian copula.

2.4 Results

2.4.1 Posterior Distributions

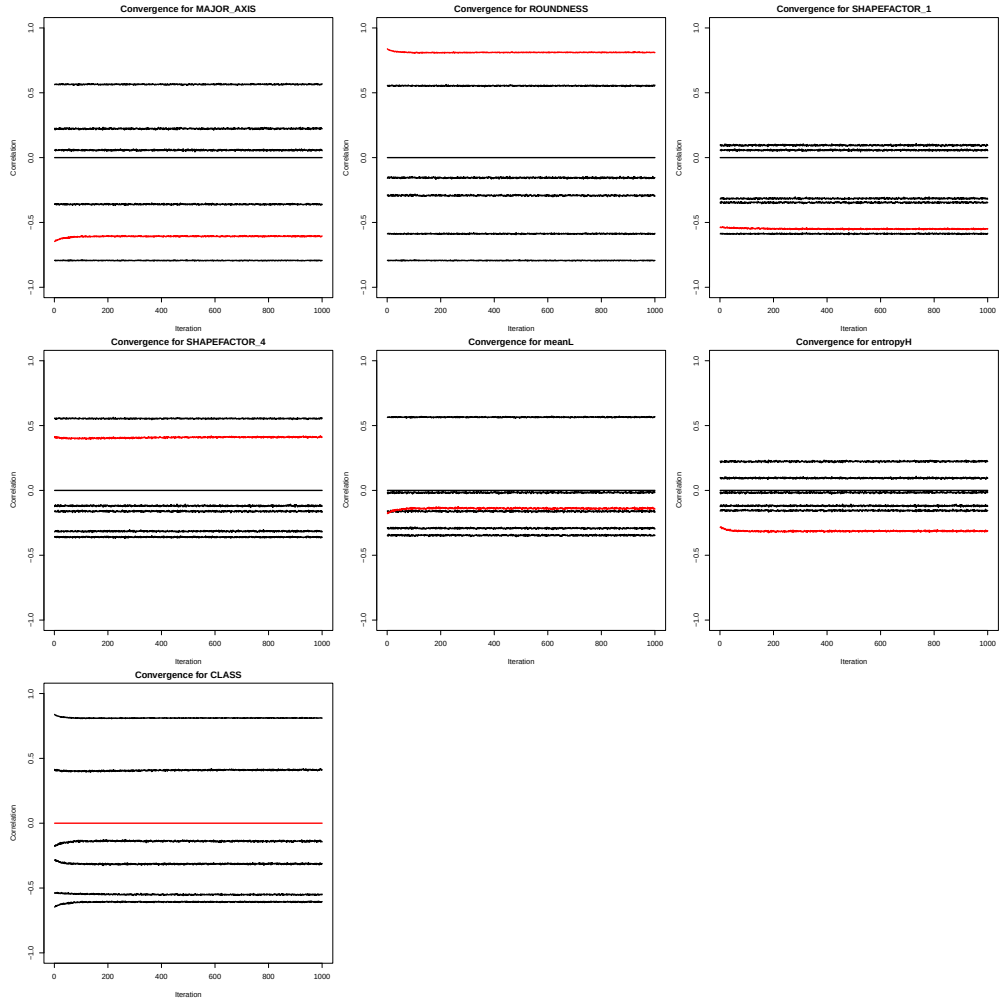

Figure 1: Trace plots of correlation samples.

Figure 1 display the posterior correlation samples for each C_i, with the plots being red when j = CLASS. Although difficult to see, the samples have successfully converged to a stationary distribution. The average effective size is 724.46. Despite the effective sizes for

all correlations that involve `CLASS` having extremely low sizes, we verify convergence with Figure 2.

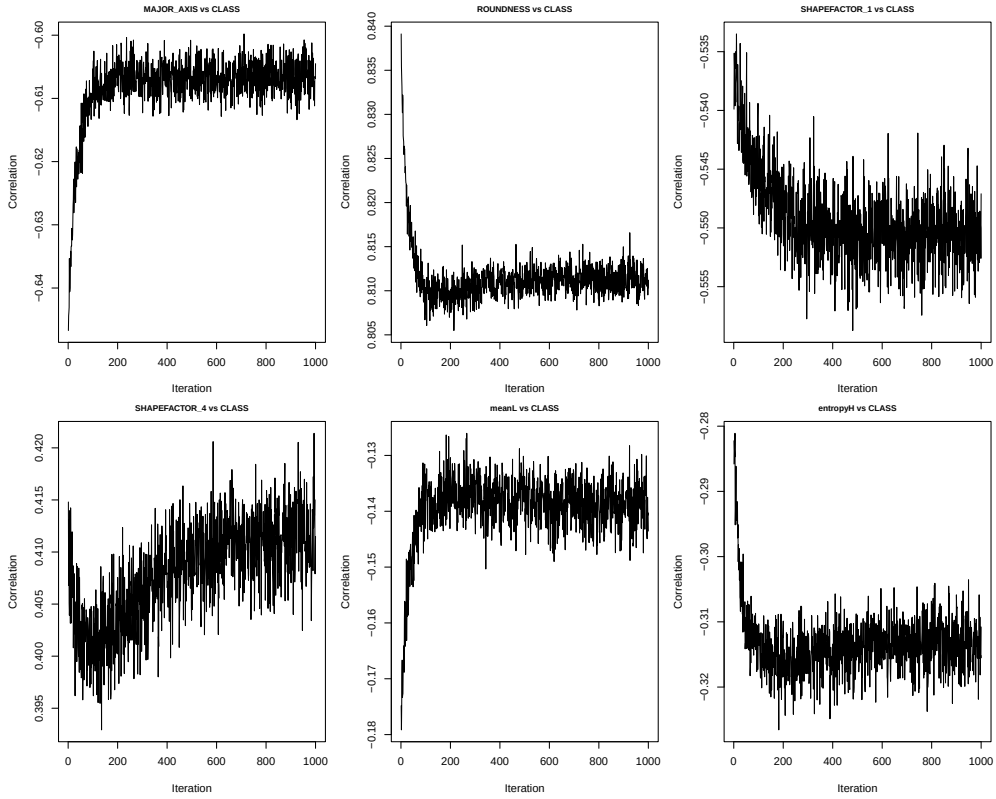

Figure 2: Trace plots of correlation samples (CLASS)

A summary of the model fit can be found in the appendix, including the effective sizes and the posterior quantiles of the correlation and regression coefficients.

2.4.2 Simulated Data

	MAJOR_AXIS	ROUNDNESS	SHAPEFACTOR_1	SHAPEFACTOR_4	meanL	entropyH	CLASS
Min.	96.97	0.3925	0.01130	0.8962	164.7	262.2	1
1st Qu.	132.62	0.6206	0.01700	0.9816	213.3	1945.0	2
Median	149.34	0.7754	0.01860	0.9864	221.6	2338.8	3
Mean	161.80	0.7325	0.02062	0.9855	222.2	2309.9	3
3rd Qu.	197.46	0.8345	0.02620	0.9907	230.2	2713.3	4
Max.	255.65	0.9800	0.03690	0.9990	252.5	4868.4	5

Table 1: Summary of the Original Data

Comparing Table 2 to Table 1, it's evident that the Gaussian copula model has managed to replicate certain attributes of the original data quite well, while some discrepancies are present in the `Min` and `Max` values.

	MAJOR_AXIS	ROUNDNESS	SHAPEFACTOR_1	SHAPEFACTOR_4	meanL	entropyH	CLASS
Min.	103.4	0.4045	0.01176	0.9470	185.9	520.9	1.000
1st Qu.	132.6	0.6316	0.01688	0.9820	213.4	1910.8	2.000
Median	146.8	0.7730	0.01860	0.9866	221.5	2272.8	3.000
Mean	159.7	0.7380	0.02045	0.9860	222.2	2255.9	3.052
3rd Qu.	195.6	0.8429	0.02610	0.9911	230.1	2685.3	4.000
Max.	236.7	0.9629	0.03248	0.9988	249.7	4388.1	5.000

Table 2: Summary of the Simulated Data

The most obvious disparities are in the tails, a characteristic limitation of the model. While the transformation ensures that $y_{i,j}$ maintains the dependency structure, it may not capture extreme tail dependencies that might be present in the data. Extreme values are less frequent than often observed in real-world data and are not be effectively mirrored in the simulated data.

The tail dependencies can be expressed as:

$$\lambda_L = \lim_{q \to 0} P[U_2 \leq q | U_1 \leq q] \tag{8}$$

$$\lambda_U = \lim_{q \to 1} P[U_2 \geq q | U_1 \geq q] \tag{9}$$

The Gaussian copula implies a symmetric tail dependence, such that:

$$\lambda_L = \lambda_U = 2\Phi(-\sqrt{\frac{\rho}{1-\rho}}) - 1 \tag{10}$$

Where ρ is an element of the correlation matrix C, describing the correlation between two variables U_1 and U_2.

For the Gaussian copula, $\lambda_L = \lambda_U = 0$ when the correlation ρ is not equal to 1, implying that it doesn't capture lower tail dependence unless the correlation is perfect [5] [6].

Figure 3 shows the side-by-side differences between variables to determine whether the Gaussian copula model adequately captures the joint tail behavior. Specifically, we focused on the 95th quantile of variable pairs, `MAJOR_AXIS` vs `ROUNDNESS`, `SHAPEFACTOR_1` vs `SHAPEFACTOR_4`, and `meanL` vs `entropyH`. As expected, the simulated data exhibits fewer data points in the upper tail regions. The simulated data does not retain the clusters and patterns inherent in the original dataset.

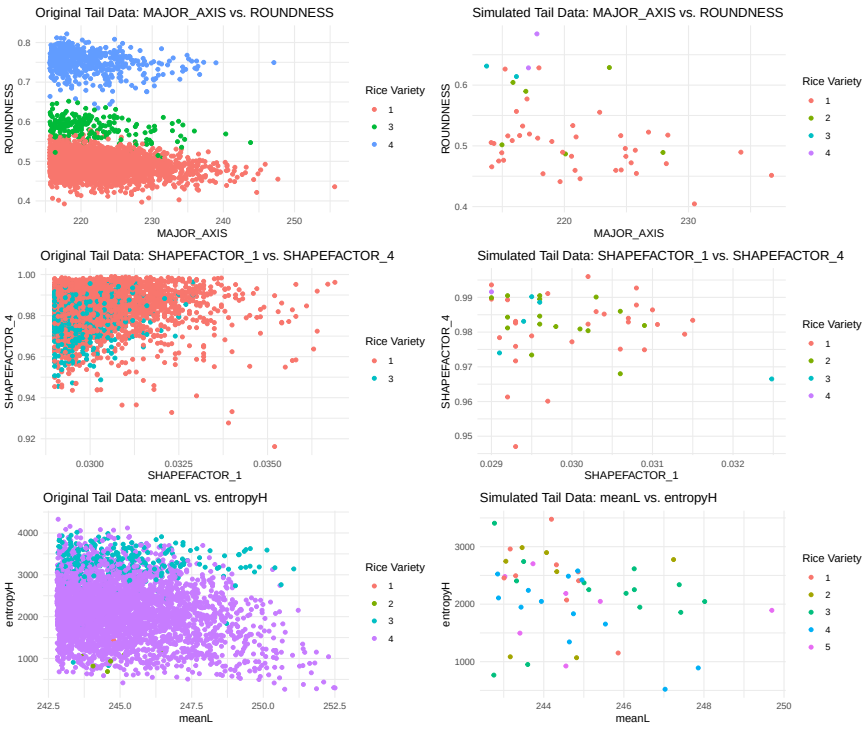

Figure 3: Comparative tail dependencies for original and simulated data

Figure 4 shows the distribution of the seven variables. The figure also shows the interrelation between those variables on the off-diagonal blocks of the matrix. We see that the distribution of variables approximates the assumed multivariate normal distribution, but with multiple modes originated from the discontinuities of the distributions of observed variables. While the data largely aligns with what one might expect from a multivariate normal distribution, the presence of multiple modes suggests some irregularities or patterns in the observed variables.

Table 3 presents the correlation matrix of the seven rice grain features. The off-diagonal blocks of the matrix distinctly highlight the pairwise interrelations between these features.

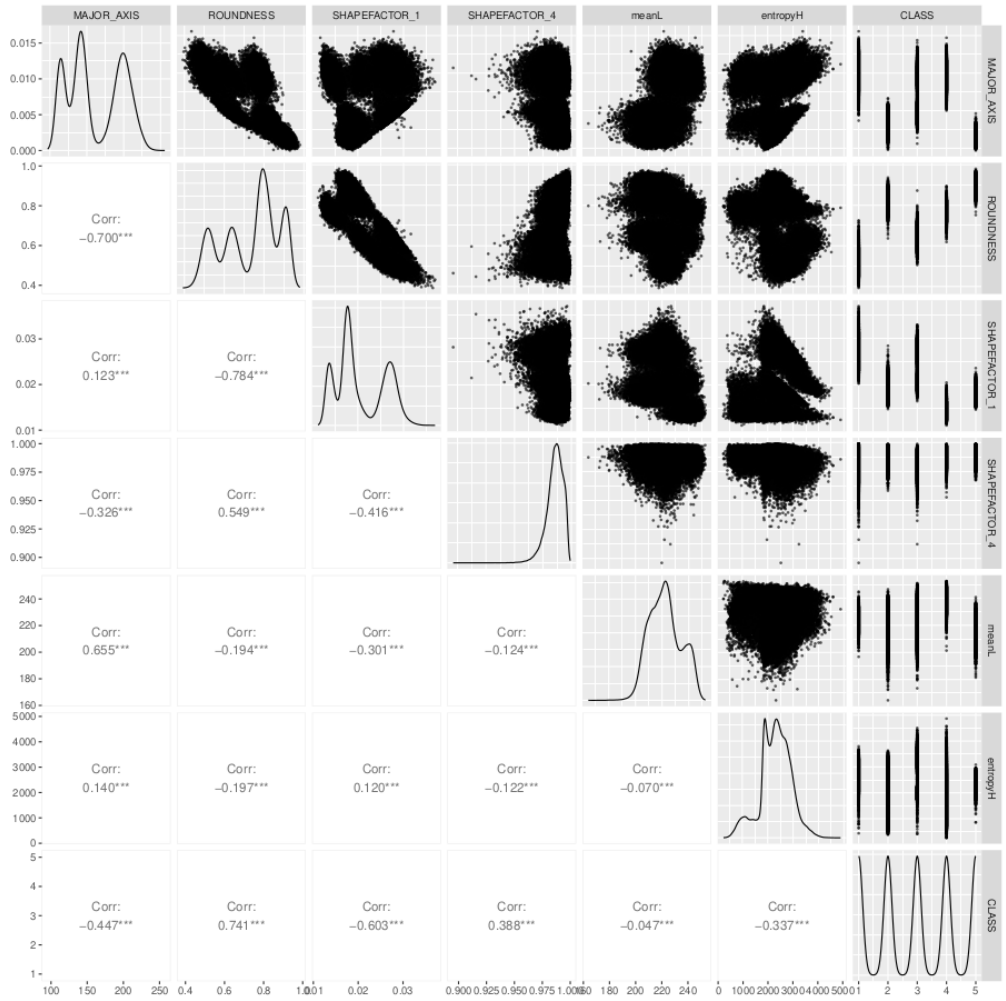

Figure 4: Distribution of the predicted latent variables

Notable observations can be inferred from Figure 4 and Table 3. The significant negative correlation of 0.793 between `MAJORAXIS` and `ROUNDNESS` indicates that as the `MAJORAXIS` length of a grain expands, its `ROUNDNESS` tends to decrease. This interaction can be attributed to the inherent shape dynamics of rice grains, where elongated grains inherently exhibit reduced roundness. A noticeable negative correlation of 0.608 between the `MAJORAXIS` and `CLASS` suggests that grains with an extended `MAJORAXIS` might be categorized differently from their shorter counterparts. The exact nature of this classification difference remains a subject for further study. There is a strong positive correlation of 0.811 between `ROUNDNESS` and `CLASS`, implying that a grain's `ROUNDNESS` could be a defining factor in its classification. The negative correlation of 0.587 between `ROUNDNESS` and `SHAPEFACTOR1` suggests that as grains become rounder, the value of `SHAPEFACTOR1` diminishes. The correlation matrix indicates a moderate negative correlation of 0.549 between `SHAPEFACTOR1` and `CLASS`. This interaction hints at a potential inverse relationship between the `SHAPEFACTOR1` of a grain and its `CLASS` classification.

	MAJOR_AXIS	ROUNDNESS	SHAPEFACTOR_1	SHAPEFACTOR_4	meanL	entropyH	CLASS
MAJOR_AXIS	1.00000000	-0.7934195	0.05737538	-0.3600173	0.56485581	0.22335041	-0.6080948
ROUNDNESS	-0.79341951	1.0000000	-0.58706813	0.5543133	-0.29203022	-0.15518655	0.8114975
SHAPEFACTOR_1	0.05737538	-0.5870681	1.00000000	-0.3153080	-0.34640881	0.09512306	-0.5491887
SHAPEFACTOR_4	-0.36001734	0.5543133	-0.31530797	1.0000000	-0.16123667	-0.11859812	0.4080429
meanL	0.56485581	-0.2920302	-0.34640881	-0.1612367	1.00000000	-0.01640562	-0.1393201
entropyH	0.22335041	-0.1551866	0.09512306	-0.1185981	-0.01640562	1.00000000	-0.3133826
CLASS	-0.60809479	0.8114975	-0.54918868	0.4080429	-0.13932007	-0.31338260	1.0000000

Table 3: Correlation Matrix for Rice Grain Features

3 Conclusions

In conclusion, our exploration into the interdependencies of rice grain features using the Gaussian copula model has showed interesting, yet expected observations. This model effectively revealed the relationships among the variables, setting a strong foundation for future work such as identification and precision methods. As expected, there is a significant connection among morphological, shape, and colour attributes. The scatter plots and correlation matrices highlighted these relationships, showing the traits that set apart different rice varieties.

The "CLASS" metric is central to rice grain categorization. The correlation between the major axis length and "CLASS" indicates that grain length influences its category. An inverse relationship exists between "SHAPEFACTOR1" and "CLASS", showing that specific shape characteristics impact classification. There is also a relationship between roundness and class, with grains of specific roundness values falling into particular categories. The "CLASS" variable reflects the collective attributes of rice grains, and understanding these relationships is crucial for refining classification models and improving rice variety identification.

Whilst it may be obvious to the reader that these variables are expected to be correlated to each other and the classification of rice, the utilization of the Gaussian copula model provides a more nuanced understanding. The Gaussian copula model allows for capturing dependencies between variables even when the marginal distributions of each variable are not necessarily Gaussian It offers flexibility in modeling multivariate distributions by decoupling the marginal distributions from the copula, which describes the dependencies.

3.1 Limitations and critique

The most obvious limitation is the computing power. We were only able to test 5% of the potential variables to answer our question of interest. There are a few reasons that I can reason this to: integration over high-dimensional spaces to determine the joint distribution, matrix operations like inversions and decompositions of the correlation matrix and the required individual transformations for each variable, particularly the computation of inverses of their respective univariate CDFs. Despite best efforts, I was unable to find a solution to these challenges.

Ideally, if I were to redo this project, I would consider employing a more targeted modeling approach. The Gaussian copula, while robust in capturing dependencies among features,

focuses on estimating the joint distribution without a specific target variable in focus. This posed a challenge when our research objective leaned towards understanding the relationship between the traits of rice and the prediction of `CLASS`. With more time, I would have applied a classification model to the simulated data to make predictions about the classification of rice. Specifically Decision Trees and Random Forests for their ability to handle non-linear relationships and interactions between variables without requiring explicit specification. Although not a Bayesian approach, I would have also been interested to apply the Semiparametric Gaussian Copula Regression Model (SGCRM) discussed by Debangan Dey and Vadim Zipunnikov [7]. This model promises a more flexible framework allowing for linear regression on the latent SGC space, offering mutually consistent conditional regression models for varied outcomes. This would have aligned well with our intended research outcomes. The SGCRM also would be computationally efficient due to being likelihood free and operating without making assumptions about the data distribution.

4 Appendix

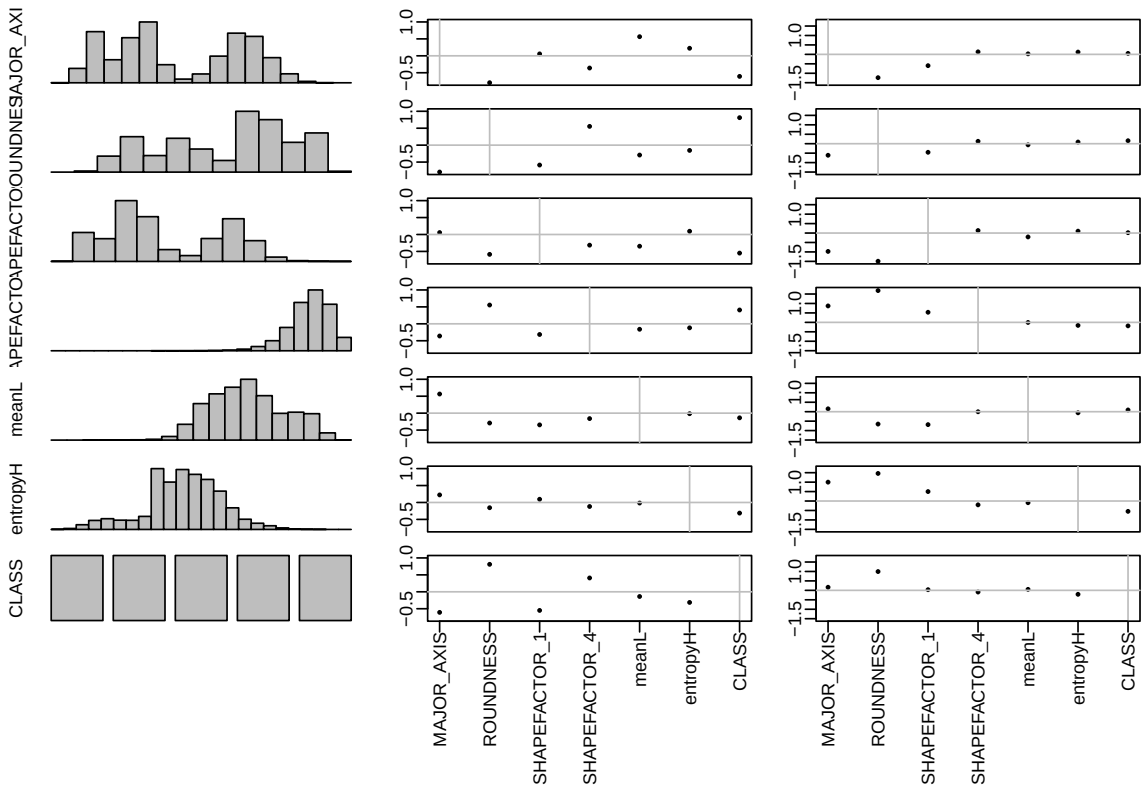

Figure 5: Summary plot of model fit

Table 4: MCMC Details

Detail	Value
number of saved samples	1000
average effective sample size	724.4594

Table 5: Effective Sample Sizes

Variable	Value
MAJOR_AXIS*ROUNDNESS	1114.09
MAJOR_AXIS*SHAPEFACTOR_1	1319.26
MAJOR_AXIS*SHAPEFACTOR_4	1006.37
MAJOR_AXIS*meanL	1128.93
MAJOR_AXIS*entropyH	605.22
MAJOR_AXIS*CLASS	22.68
ROUNDNESS*SHAPEFACTOR_1	681.79
ROUNDNESS*SHAPEFACTOR_4	882.26
ROUNDNESS*meanL	1001.09
ROUNDNESS*entropyH	882.20
ROUNDNESS*CLASS	27.83
SHAPEFACTOR_1*SHAPEFACTOR_4	828.99
SHAPEFACTOR_1*meanL	1694.13
SHAPEFACTOR_1*entropyH	921.44
SHAPEFACTOR_1*CLASS	24.61

Table 6: Posterior Quantiles of Correlation Coefficients

Variable	2.5% quantile	50% quantile	97.5% quantile
MAJOR_AXIS*ROUNDNESS	-0.80	-0.79	-0.79
MAJOR_AXIS*SHAPEFACTOR_1	0.05	0.06	0.06
MAJOR_AXIS*SHAPEFACTOR_4	-0.37	-0.36	-0.35
MAJOR_AXIS*meanL	0.56	0.56	0.57
MAJOR_AXIS*entropyH	0.22	0.22	0.23
MAJOR_AXIS*CLASS	-0.62	-0.61	-0.60
ROUNDNESS*SHAPEFACTOR_1	-0.59	-0.59	-0.58
ROUNDNESS*SHAPEFACTOR_4	0.55	0.55	0.56
ROUNDNESS*meanL	-0.30	-0.29	-0.29
ROUNDNESS*entropyH	-0.16	-0.16	-0.15
ROUNDNESS*CLASS	0.81	0.81	0.82
SHAPEFACTOR_1*SHAPEFACTOR_4	-0.32	-0.32	-0.31
SHAPEFACTOR_1*meanL	-0.35	-0.35	-0.34
SHAPEFACTOR_1*entropyH	0.09	0.10	0.10
SHAPEFACTOR_1*CLASS	-0.56	-0.55	-0.54
SHAPEFACTOR_4*meanL	-0.17	-0.16	-0.15
SHAPEFACTOR_4*entropyH	-0.13	-0.12	-0.11
SHAPEFACTOR_4*CLASS	0.40	0.41	0.42
meanL*entropyH	-0.02	-0.02	-0.01
meanL*CLASS	-0.16	-0.14	-0.13
entropyH*CLASS	-0.32	-0.31	-0.30

Table 7: Posterior Quantiles of Regression Coefficients

Variable	2.5% quantile	50% quantile	97.5% quantile
MAJOR_AXIS ROUNDNESS	-1.24	-1.23	-1.23
MAJOR_AXIS SHAPEFACTOR_1	-0.60	-0.60	-0.59
MAJOR_AXIS SHAPEFACTOR_4	0.13	0.13	0.14
MAJOR_AXIS meanL	0.03	0.03	0.03
MAJOR_AXIS entropyH	0.12	0.12	0.12
MAJOR_AXIS CLASS	0.05	0.05	0.06
ROUNDNESS SHAPEFACTOR_1	-0.10	-0.09	-0.09
ROUNDNESS SHAPEFACTOR_4	-0.07	-0.06	-0.06
ROUNDNESS meanL	0.02	0.02	0.02
ROUNDNESS entropyH	-0.06	-0.06	-0.06
ROUNDNESS CLASS	0.01	0.02	0.02
SHAPEFACTOR_1 SHAPEFACTOR_4	0.03	0.03	0.03
SHAPEFACTOR_1 meanL	-0.05	-0.05	-0.05
SHAPEFACTOR_1 entropyH	-0.02	-0.01	-0.01
SHAPEFACTOR_1 CLASS	0.05	0.05	0.06
SHAPEFACTOR_4 meanL	-0.06	-0.06	-0.06
SHAPEFACTOR_4 entropyH	-0.08	-0.07	-0.07
SHAPEFACTOR_4 CLASS	-0.10	-0.09	-0.09
meanL entropyH	0.01	0.02	0.02
meanL CLASS	0.05	0.06	0.06
entropyH CLASS	0.01	0.01	0.01

4.1 R Code

```
###MODEL
library(farff)

data <- readARFF("Rice_MSC_Dataset.arff")

print(unique(data$CLASS))
data <- na.omit(data)
data1 <- data
data$CLASS  <- as.numeric(factor(data$CLASS))
data1$CLASS <- as.numeric(factor(data$CLASS))

selected_vars <- c( "MAJOR_AXIS",
                    "ROUNDNESS", "SHAPEFACTOR_1", "SHAPEFACTOR_4", "meanL"
   , "entropyH", "CLASS")
data <- data[, c(selected_vars)]
data1 <- data1[, c(selected_vars)]


library(ggplot2)
library(MASS)
```

```r
library(sbgcop)

set.seed(123)
fit <- sbgcop.mcmc(data, S0 = diag(ncol(data)), nsamp = 75000)
summary(fit)
plot(fit)

library(mvtnorm)

n_sims <- dim(fit$C.psamp)[3]
simulations <- array(NA, dim = c(n_sims, ncol(data)))


for (i in 1:n_sims) {

  sampled_corr_matrix <- fit$C.psamp[,,i]

  z <- rmvnorm(1, sigma = sampled_corr_matrix)

  for (j in 1:ncol(data)) {
    simulations[i, j] <- quantile(data[, j], probs = pnorm(z[j]), na.rm =
   TRUE, type = 8)
  }
}

simulated_data <- as.data.frame(simulations)
colnames(simulated_data) <- colnames(data)



original_summary <- summary(data[, selected_vars])
simulated_summary <- summary(simulated_data[, selected_vars])

###VISUALISATION

library(knitr)

cat("Original Data Summary\n")
kable(original_summary)

cat("Simulated Data Summary\n")
kable(simulated_summary)

# Extract posterior samples of the correlation matrix
correlation_samples <- fit$C.psamp
mean_correlations <- apply(correlation_samples, c(1,2), mean)
print(mean_correlations)

sample_corr <- correlation_samples[,,50]

library(GGally)
pdf("scatter_gc.pdf", width = 12, height = 12)

ggpairs(data, upper = list(continuous = wrap("points", alpha = 0.5, size =
```

```
    0.5)),
        lower = list(continuous = "cor"))
dev.off()


n_vars <- dim(correlation_samples)[1]
n_combinations <- choose(n_vars, 2)
n_rows <- 2
n_cols <- 5
vars <- colnames(data)

plot_pairs_class <- function(start, end) {
  class_index <- length(vars)
  plot_count <- start - 1
  for (i in 1:(class_index-1)) {
    plot_count <- plot_count + 1
    if (plot_count >= start && plot_count <= end) {
      if(vars[i] != "CLASS") {
        plot(correlation_samples[i, class_index, ], type="l", col="black",
    ylab="Correlation", xlab="Iteration", main=paste(vars[i], "vs CLASS"),
    cex.main=0.8)
      }
    }
    if (plot_count == end) {
      return()
    }
  }
}

pdf("tp_gc1.pdf", width=10, height=8)
par(mfrow=c(2, 3), mar=c(4, 4, 2, 1))
plot_pairs_class(1, 10)
dev.off()
par(mfrow=c(1, 1), mar=c(5, 4, 4, 2))


# Correlation convergence trace plots
prepare_matrix_for_var <- function(var_index) {
  matrix_data <- matrix(0, nrow=dim(correlation_samples)[3], ncol=n_vars)

  for (i in 1:n_vars) {
    if (i != var_index) {
      matrix_data[, i] <- correlation_samples[var_index, i, ]
    }
  }

  return(matrix_data)
}
overlay_correlation_for_variable <- function(var_index) {
  mat_data <- prepare_matrix_for_var(var_index)
  matplot(mat_data, type="l", lty=1,
          xlab="Iteration", ylab="Correlation",
          main=paste("Convergence for", vars[var_index]),
          col=ifelse(vars == "CLASS", "red", "black"),
```

```
          ylim=c(-1, 1), lwd=2)
}
layout_dim <- ceiling(sqrt(n_vars))
pdf("correlation_convergence_overlays.pdf", width=15, height=15)
par(mfrow=c(layout_dim, layout_dim), mar=c(4, 4, 2, 1))
for (i in 1:n_vars) {
  overlay_correlation_for_variable(i)
}
par(mfrow=c(1, 1), mar=c(5, 4, 4, 2))
dev.off()

#Original v Simulated Scatterplots
pdf("scatter_comparison.pdf", width = 20, height = 15)
p1 <- ggplot(data1, aes(x = MAJOR_AXIS, y = ROUNDNESS, color = factor(
    CLASS))) +
  geom_point() +
  theme_minimal() +
  labs(title = "Original Data", color = "Rice Variety")

p2 <- ggplot(simulated_data, aes(x = MAJOR_AXIS, y = ROUNDNESS, color =
    factor(CLASS))) +
  geom_point() +
  theme_minimal() +
  labs(title = "Simulated Data", color = "Rice Variety")


p3 <- ggplot(data1, aes(x = SHAPEFACTOR_1, y = SHAPEFACTOR_4, color =
    factor(CLASS))) +
  geom_point() +
  theme_minimal() +
  labs(color = "Rice Variety")

p4 <- ggplot(simulated_data, aes(x = SHAPEFACTOR_1, y = SHAPEFACTOR_4,
    color = factor(CLASS))) +
  geom_point() +
  theme_minimal() +
  labs(title = "Simulated Data", color = "Rice Variety")

p5 <- ggplot(data1, aes(x = meanL, y = entropyH, color = factor(CLASS))) +
  geom_point() +
  theme_minimal() +
  labs(color = "Rice Variety")

p6 <- ggplot(simulated_data, aes(x = meanL, y = entropyH, color = factor(
    CLASS))) +
  geom_point() +
  theme_minimal() +
  labs(title = "Simulated Data", color = "Rice Variety")
grid.arrange(p1, p2, p3, p4, p5, p6, ncol = 2)
dev.off()


library(ggplot2)
library(gridExtra)
```

```r

quantile_threshold <- 0.95
pdf("tail_dependencies_comparison.pdf", width = 12, height = 10)
tail_data <- data1[which(data1$MAJOR_AXIS > quantile(data1$MAJOR_AXIS,
    quantile_threshold)), ]
tail_simulated_data <- simulated_data[which(simulated_data$MAJOR_AXIS >
    quantile(simulated_data$MAJOR_AXIS, quantile_threshold)), ]

p1 <- ggplot(tail_data, aes(x = MAJOR_AXIS, y = ROUNDNESS, color = factor(
    CLASS))) +
  geom_point() +
  theme_minimal() +
  labs(title = "Original Tail Data: MAJOR_AXIS vs. ROUNDNESS", color = "
    Rice Variety")

p2 <- ggplot(tail_simulated_data, aes(x = MAJOR_AXIS, y = ROUNDNESS, color
     = factor(CLASS))) +
  geom_point() +
  theme_minimal() +
  labs(title = "Simulated Tail Data: MAJOR_AXIS vs. ROUNDNESS", color = "
    Rice Variety")

tail_data <- data1[which(data1$SHAPEFACTOR_1 > quantile(data1$SHAPEFACTOR_
    1, quantile_threshold)), ]
tail_simulated_data <- simulated_data[which(simulated_data$SHAPEFACTOR_1 >
     quantile(simulated_data$SHAPEFACTOR_1, quantile_threshold)), ]

p3 <- ggplot(tail_data, aes(x = SHAPEFACTOR_1, y = SHAPEFACTOR_4, color =
    factor(CLASS))) +
  geom_point() +
  theme_minimal() +
  labs(title = "Original Tail Data: SHAPEFACTOR_1 vs. SHAPEFACTOR_4",
    color = "Rice Variety")

p4 <- ggplot(tail_simulated_data, aes(x = SHAPEFACTOR_1, y = SHAPEFACTOR_
    4, color = factor(CLASS))) +
  geom_point() +
  theme_minimal() +
  labs(title = "Simulated Tail Data: SHAPEFACTOR_1 vs. SHAPEFACTOR_4",
    color = "Rice Variety")

tail_data <- data1[which(data1$meanL > quantile(data1$meanL, quantile_
    threshold)), ]
tail_simulated_data <- simulated_data[which(simulated_data$meanL >
    quantile(simulated_data$meanL, quantile_threshold)), ]

p5 <- ggplot(tail_data, aes(x = meanL, y = entropyH, color = factor(CLASS)
    )) +
  geom_point() +
  theme_minimal() +
  labs(title = "Original Tail Data: meanL vs. entropyH", color = "Rice
    Variety")

p6 <- ggplot(tail_simulated_data, aes(x = meanL, y = entropyH, color =
```

```
    factor(CLASS))) +
  geom_point() +
  theme_minimal() +
  labs(title = "Simulated Tail Data: meanL vs. entropyH", color = "Rice
    Variety")

grid.arrange(p1, p2, p3, p4, p5, p6, ncol = 2)
dev.off()


correlation_samples <- fit$C.psamp
mean_correlations <- apply(correlation_samples, c(1,2), mean)
print(mean_correlations)
```

References

[1] Murat Koklu. *Rice MSC Dataset*. URL: https://www.muratkoklu.com/datasets/.

[2] İlkay Çınar and Murat Köklü. "Determination of Effective and Specific Physical Features of Rice Varieties by Computer Vision In Exterior Quality Inspection". In: *Selcuk Journal of Agriculture and Food Sciences* 35.3 (2021). URL: https://dergipark.org.tr/en/pub/selcukjafsci/issue/76652/1277427#article_cite.

[3] P.D. Hoff. "Extending the rank likelihood for semiparametric copula estimation". In: *Ann. Appl. Stat.* 1.1 (2007), pp. 265–283.

[4] P.D. Hoff. *sbgcop: Semiparametric Bayesian Gaussian copula estimation and imputation*. 2007. URL: https://github.com/pdhoff/sbgcop.

[5] A. Ruckstuhl. *Statistical Analysis of Financial Data: Lecture 5*. Lecture notes. ETH Zürich & Zürcher Hochschule für Angewandte Wissenschaften (ZHAW), Institut für Datenanalyse und Prozess Design IDP, 2021.

[6] A. AghaKouchak, S. Sellars, and S. Sorooshian. *Methods of Tail Dependence Estimation*. URL: https://escholarship.org/content/qt07x6p3bk/qt07x6p3bk_noSplash_c6faf62d3de3c34d81607b2465a48c15.pdf?t=q8hduy.

[7] D. Dey and V. Zipunnikov. *Semiparametric Gaussian Copula Regression modeling for Mixed Data Types (SGCRM)*. 2022. URL: https://arxiv.org/abs/2205.06868.