UNIVERSIDAD
EAFIT®

Vigilada Mineducación

Nonparametric Generation of Synthetic Data Using Copulas

Juan Pablo Restrepo Lopera

Tesis de Maestría

Asesor
Henry Laniado Rodas

Co-asesor
Juan Carlos Rivera Agudelo

UNIVERSIDAD EAFIT
ESCUELA DE CIENCIAS
MAESTRÍA EN MATEMÁTICAS APLICADAS
MEDELLÍN
2023

electronics

MDPI

Article

Nonparametric Generation of Synthetic Data Using Copulas

Juan P. Restrepo , Juan Carlos Rivera , Henry Laniado *, Pablo Osorio and Omar A. Becerra

Mathematical Modeling Research Group, Universidad EAFIT, Medellin 050022, Colombia
* Correspondence: hlaniado@eafit.edu.co

Abstract: This article presents a novel nonparametric approach to generate synthetic data using copulas, which are functions that explain the dependency structure of the real data. The proposed method addresses several challenges faced by existing synthetic data generation techniques, such as the preservation of complex multivariate structures presented in real data. By using all the information from real data and verifying that the generated synthetic data follows the same behavior as the real data under homogeneity tests, our method is a significant improvement over existing techniques. Our method is easy to implement and interpret, making it a valuable tool for solving class imbalance problems in machine learning models, improving the generalization capabilities of deep learning models, and anonymizing information in finance and healthcare domains, among other applications.

Keywords: synthetic data generation; data augmentation; homogeneity test; empirical copulas; nonparametric statistics

check for updates

Citation: Restrepo, J.P.; Rivera, J.C.; Laniado, H.; Osorio, P.; Becerra, O. Nonparametric Generation of Synthetic Data Using Copulas. *Electronics* **2023**, *12*, 1601. https://doi.org/10.3390/electronics12071601

Academic Editors: Juan M. Corchado and Gorka Epelde Unanue

Received: 31 January 2023
Revised: 15 March 2023
Accepted: 16 March 2023
Published: 29 March 2023

1. Introduction

Synthetic data (**SD**) is data generated through mathematical models that preserve the statistical properties of real data (**RD**), such as the marginal and joint distributions of the data variables [1,2]. In recent years, research on synthetic data generation processes has become more relevant since important applications have been demonstrated, such as the possibility of anonymizing information, with a special interest in health [3–5], balancing classes in the training of machine learning (**ML**) models [6–9], increasing the amount of data to improve the generalizability of deep learning models [10–12], among others.

The generation of **SD** makes it possible to solve the problem of class imbalance in the **ML** algorithms used in classification, thanks to the fact that the **RD** are oversampled, which allows for obtaining new individuals from minority classes. One of the most widely used techniques for this has been the Synthetic Minority Oversampling Technique (**SMOTE**). Originally introduced by Chawla et al. [13], **SMOTE** finds the nearest neighbors for a random sample of the class of interest and then randomly selects one of those neighbors and generates a sample that belongs to the line segment joining the random sample with its neighbor. The main disadvantage of this method is that it uses local information for data synthesis instead of considering the complete distribution of minority classes [6]. Applications of this method on the performance of **ML** models can be found in [8,9].

A novel method for generating **SD** is Generative Adversarial Networks (**GAN**), initially introduced by Goodfellow et al. [14]. **GAN** consists of coupling two neural network architectures; one of them receives the name of Generator and the other the name of Discriminator. The first one has the function of generating **SD** from the **RD** and the second one of classifying if the data generated are real or synthetic. The ultimate goal is that the synthetic samples have such good quality that they are indistinguishable for the Discriminator [10]. Although it has been a short time since they were introduced in 2014, **GAN** has been remarkably improved, to the point that even for a human being, it can be difficult to distinguish between real and synthetic images generated by the method. Their main drawback is that they are challenging to train [15]. Numerous studies present applications of **GAN**; for instance, in Porcu et al. [11], they are used to improve the generalization capacity of facial recognition **ML** models. However, in Andreini et al. [12], they are used to generate retinal images. In Poudevigne-Durance et al. [16], a modification is presented that allows the **GAN** to better manage records with missing data, among others.

Another relevant method for the generation of **SD** is the use of copulas; these are multivariate distribution functions that can explain the dependency relationships among the variables of a data set [17]. Recent work on generating **SD** from copulas is found in Patki et al. [18], where Gaussian copulas are used to generate **SD** in the context of a relational database. On the other hand, Sun et al. [18], employ vine copulas to produce **SD** that can be used to fit **ML** models. In turn, Nejad et al. [19] use Archimedean copulas to generate a synthetic population and thus carry out an emergency planning study. Despite previous references offer effective methods for generating **SD**, they share a common limitation in that they are based on parametric versions of copula theory. Explicitly, these methods assume a specific functional shape for the copula of the data under study. Although this assumption is prevalent in the literature [20–22], it can be problematic since it imposes certain restrictions on the copula that may not hold true in practice. Furthermore, it can be challenging to verify the validity of such an assumption, and therefore a wrong selection of the parametric copula could lead to distorted results.

Other examples of **SD** generation can be found in Reiter [23], who uses classification and regression trees for synthesis, Ping et al. [24], who use Bayesian neural networks, Rankin et al. [3], who apply both previous methods to preserve the privacy of data on the health status of patients, and Yale et al. [4], who fit a multivariate Gaussian distribution by maximum likelihood to generate new data, among others.

SD generation methods have proven their practical utility in a wide range of scenarios. For instance, Wang et al. [25] used synthetic data to address the challenge of insufficient data for training machine learning models in crowd analysis. In the work of Boikov et al. [26], synthetic data were used in the automated recognition of defective parts in steel production, which allowed the training of two deep learning models: one for classification and one for segmentation. Shamsolmoali et al. [27] used synthetic data to improve the generalizability of a road segmentation model. Farajzadeh-Zanjani et al. [28] used synthetic data to address the class imbalance problem and improve the training of attack detection models in electrical networks. These examples demonstrate the utility of synthetic data to address various challenges in machine learning and artificial intelligence.

In scientific literature, whenever techniques for data augmentation are introduced, authors usually present statistical tests to evaluate whether the **SD** follows the same distribution as the **RD**. Examples of the above can be found in Yale et al. [4], who verify if the confidence intervals of each simulated and real variable overlap and check if the PCA projections of **RD** and **SD** are similar. Hernandez et al. [29] use the Student's *t*-test to verify the equality of the means of the variables and the Kolmogorov–Smirnov test, which checks whether the marginal distributions of each variable are equal. Gonzalez-Abril et al. [30] use Pearson's chi-square test to verify categorical variables and the Kolmogorov–Smirnov test for continuous variables, among others.

There are review articles, such as those by Dankar et al. [31] and Hernandez et al. [32], about the standard methods to compare if the **SD** follows the exact behavior of the **RD**. However, to our knowledge, there are no articles where it is verified, using a multivariate homogeneity test, if the **SD** has the same distribution as the **RD**. For the authors of this work, it represents a great opportunity since a multivariate homogeneity test is the most appropriate way to verify that the marginal distributions, the joint distributions, and the dependency relationships (linear and nonlinear) of the **SD** are the same as those of the **RD**. This is because comparing each variable marginally is not enough to verify that the **SD** distribution is equal to the **RD** distribution since, although the marginal distributions may be equal, this does not imply that the joint distributions are. An example of the latter can be found in the work of Matejka and Fitzmaurice [33], who generate different bivariate datasets, all with the same arithmetic means, standard deviations, and Pearson correlation coefficients but with different dependency structures [34]. Figure 1 presents some of the samples generated by these authors.

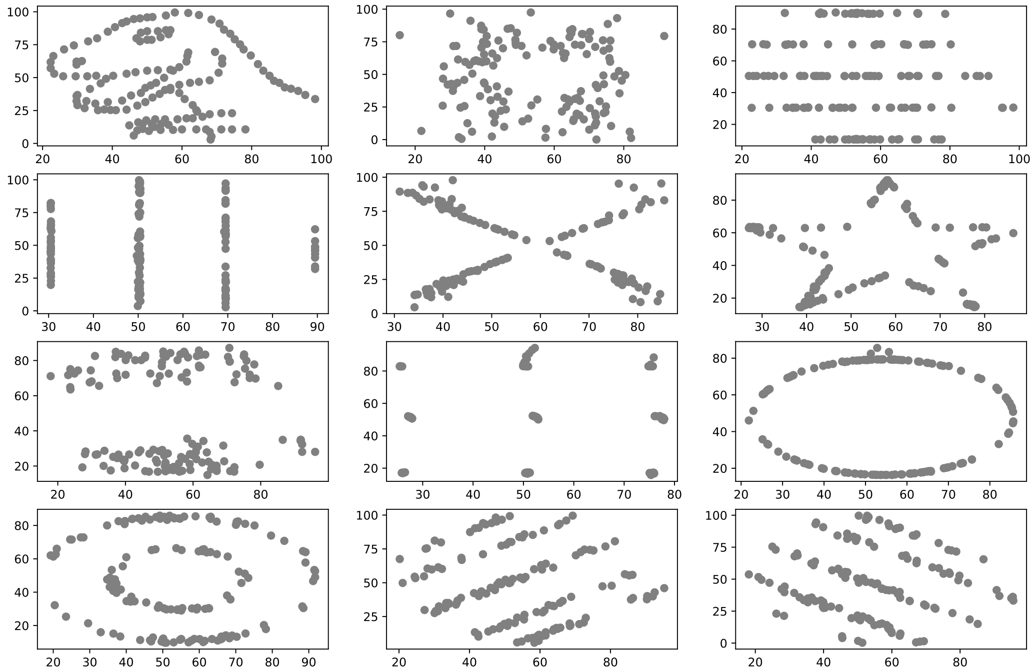


Figure 1. Same Mean, Standard Deviation and Pearson's Correlation Coefficient but Different Dependency Structure, adapted from [33,34].

This article presents a novel nonparametric method to generate **SD** through copulas, which are functions that explain the dependency structure of the **RD** [35] . The proposed method uses the global information of the data distribution, and not only local information such as **SMOTE** , which allows for fitting the dependency relationships that exist in a data set better. One of the advantages of the proposed method is that, since it is nonparametric, empirical copulas are used, which avoids having to make assumptions about the parametric distribution that the **RD** follows. In addition, this method is easier to use, interpret and implement than **GAN,** since the algorithm introduced here carries out simple calculations while achieving excellent results.

Furthermore, the multivariate homogeneity test introduced by Liu et al. [36] and known as the Data-Depth plot (**DD-plot**) is used to evaluate that the **SD** comes from the same generating process as the **RD**. This is an outstanding contribution since, in the scientific literature, many articles focus on comparing marginal distributions of **SD** with **RD** and on examining linear correlation coefficients, among other techniques. However, these measurements are not enough since they do not take into account the equality of the multivariate joint distributions. Our study fills this gap by conducting a comprehensive evaluation of the quality of the **SD**.

In summary, this article presents a new and easy-to-implement method to generate **SD** and demonstrates that the data generated by the method respects the dependency structures of **RD**, without the need to know the functional shapes of such structures. The article is organized as follows: Section 2 presents the formal development of the method, describes the simulations carried out to exemplify the method, explains the **DD-plot** and describes a sensitivity analysis performed on the algorithm. Section 3 presents the results of the simulations and analyzes the goodness of the method. Finally, Section 4 presents the conclusions.

2. Methodology

This section describes the methodology followed. Section 2.1 presents the mathematical formalization of the method, Section 2.2 describes the experiments carried out to demonstrate the goodness of the method, Section 2.3 describes the multivariate homogeneity test used, and Section 2.4 explains how a sensitivity analysis of the method was carried out.

2.1. Mathematical Framework

Consider a p-dimensional random vector $\mathcal{X} = [X_1, \ldots, X_p]$ that comes from a multivariate distribution F:

$$F(x_1, \ldots, x_p) = P(X_1 \leq x_1, \ldots, X_k \leq x_p).$$

Let F_i be the marginal distribution of random variable X_i, $F_i(x) = P(X_i \leq x)$. Now, recall that, if $U \sim$ uniform$[0,\ 1]$, then $F_U(u) = P(U \leq u) = u$. From here, one can conclude that $F_i(X_i) \sim$ uniform$[0,\ 1]$, since

$$P(F_i(X_i) \leq u) = P(X_i \leq F_i^{-1}(u)) = F_i(F_i^{-1}(u)) = u. \tag{1}$$

It is worth pointing out that the previous result is quite known and valuable for generating synthetic data from the random variable X_i with distribution function F_i. It is a simple step by generating a random variable $U \sim$ uniform$[0,\ 1]$ and then to consider $X_i = F_i^{-1}(U) = \inf\{X \mid F_i(X) \geq U\}$; here, F_i must be known. However, to generate a sample of synthetic data of size N from observed random sample $X_{1i}, \cdots, X_{ni}$ coming from the same distribution function F_i where F_i is unknown, one must implement some nonparametric simulation process. In this paper, we introduce a methodology to generate synthetic data that must come from the same population where the observed sample $X_{1i}, \cdots, X_{ni}$ is coming.

Let $X_{[r]i}$ be the r-th order statistic of a random sample $X_{1i}, \cdots, X_{ni}$, i.e.,

$$X_{[1]i} \leq X_{[2]i} \leq \cdots \leq X_{[n]i}.$$

Consider a partition of the interval $[X_{[1]i},\ X_{[n]i}]$ as $X_{[1]i} = a_0 < a_1 < \cdots < a_t = X_{[n]i}$. Define B_s as:

$$B_s = \begin{cases} [a_{s-1},\ a_s] & \text{if } s = 1 \\ (a_{s-1},\ a_s] & \text{otherwise} \end{cases}$$

$$R(B_s) = \frac{1}{n} \sum_{j=1}^{s} \sum_{k=1}^{n} I_{\{X_{ki} \in B_j\}}, \quad \forall\, s \in \{1, \cdots, t\}, \tag{2}$$

where $I_{\{\cdot\}}$ is the indicator function. Note that $R(B_s)$ can be seen as a natural estimator for $F_i(a_s)$, thus some desirable properties will be considered below. In the context of density estimation, an important parameter to consider is the bandwidth denoted by h, which represents the radius of each element in the partition, i.e., $h = (a_j - a_{j-1})/2$. It is worth noting that the number of partitions t in Equation (2) depends on the bandwidth parameter as follows: $t = (X_{[n]i} - X_{[1]i})/(2h)$. Some methods for computing the bandwidth h are discussed in greater detail in Wasserman ([37], pp. 134–135).

Proposition 1. *$R(B_s)$ is an unbiased estimator for $F_i(a_s)$*

Proof.

$$E[R(B_s)] = E\left[\frac{1}{n}\sum_{j=1}^{s}\sum_{k=1}^{n} I_{\{X_{ki}\in B_j\}}\right] = \frac{1}{n}\sum_{j=1}^{s}\sum_{k=1}^{n} E\left[I_{\{X_{ki}\in B_j\}}\right] = \frac{1}{n}\sum_{j=1}^{s}\sum_{k=1}^{n} P\left(X_{ki}\in B_j\right)$$

$$= \frac{1}{n}\sum_{j=1}^{s} n\left(F_i(a_j) - F_i(a_{j-1})\right) = \sum_{j=1}^{s}\left(F_i(a_j) - F_i(a_{j-1})\right) = -F_i(a_0) + F_i(a_s).$$

However, $F_i(a_0) = P\left(X_i < X_{[1]i}\right) = 0$; then, the proof is completed. □

Previous results of Proposition 1 will be needed in the proof of the next proposition that states an asymptotic result.

Proposition 2. *$R(B_s)$ converges in quadratic mean to $F_i(a_s)$*

Proof.

$$V[R(B_s)] = V\left[\frac{1}{n}\sum_{j=1}^{s}\sum_{k=1}^{n} I_{\{X_{ki}\in B_j\}}\right] = \frac{1}{n^2}\sum_{j=1}^{s}\sum_{k=1}^{n} V\left[I_{\{X_{ki}\in B_j\}}\right]$$

$$= \frac{1}{n^2}\sum_{j=1}^{s}\sum_{k=1}^{n} P\left(X_{ki}\in B_j\right)\left[1 - P\left(X_{ki}\in B_j\right)\right]$$

$$= \frac{1}{n^2}\sum_{j=1}^{s} n\left(F_i(a_j) - F_i(a_{j-1})\right)\left(1 - F_i(a_j) + F_i(a_{j-1})\right)$$

$$= \frac{1}{n}\sum_{j=1}^{s}\left(F_i(a_j) - F_i(a_{j-1})\right)\left(1 - F_i(a_j) + F_i(a_{j-1})\right) \le \frac{s}{n}.$$

Therefore, $V[R(B_s)]$ converges to zero, and from Proposition 1, we have the desired conclusion. □

It is recalled that, if one wants to generate univariate synthetic data from a known distribution $F_i(x)$, a simple form is using the inverse transform, i.e., one must generate a random variable Uniform in $[0,\ 1]$ and then $X_i = F_i^{-1}(U) = \inf\{X \mid F_i(X) \ge U\}$. However, if $F_i(x)$ is unknown but we have a sample $X_{1i}, \cdots, X_{ni}$ that comes from the distribution $F_i(x)$, the generation of synthetic data is more complicated. In this paper, considering the result obtained in Proposition 2, a methodology is proposed to generate univariate synthetic data but when one has a sample $X_{1i}, \cdots, X_{ni}$ that comes from the unknown distribution $F_i(x)$. Following the same idea considered in the inverse transform, one must generate a random variable U Uniform in $[0,\ 1]$ and then calculate $\min\{s \mid R(B_s) \ge U\}$ and so the synthetic data $\hat{X}$ are generated by considering $\hat{X} = a_{s-1} + (a_s - a_{s-1})U$.

Example 1. *Let us simulate 300 data points from a standard Normal distribution. These 300 data points will be considered as an observed sample, i.e., raw data. Now, from those raw data, we will generate 1000 synthetic data points through the procedure introduced in this paper. The results on some relevant parameters can be seen in Table 1.*

We will generate 2000 collections of synthetic data, each comprising 1000 records and based on the same raw data that were previously considered. For each data set, the same values considered in Table 1 are calculated. Table 2 displays the mean value and the 95% confidence interval of each of those values.

Figure 2 shows the curve of theoretical distribution, the empirical distribution of raw data and the empirical distribution of synthetic data. Note that the raw data curve fits well to the theoretical distribution curve. This is not surprising because the raw data have been simulated from the theoretical distribution. However, the synthetic data generated from the raw data retain the same *good-fit behavior. Note that the quality of the synthetic data depends on the quality of the observed raw data.*

Table 1. Population parameters of a standard normal distribution and their point estimates considering the raw data and synthetic data generated from those raw data.

Standard Normal	P_{25}	Mean	Std	Median	P_{75}
Theoretical Value	−0.6745	0	1	0	0.6745
Raw Data	−0.6504	−0.0023	0.9998	−0.0186	0.6368
Synthetic Data	−0.6425	−0.0028	1.0054	−0.0119	0.6502

Table 2. Mean value and confidence intervals for the 2000 synthetic data collections.

Standard Normal	P_{25}	Mean	Std	Median	P_{75}
Mean Value	−0.6522	−0.0027	1.0026	−0.0135	0.6389
Confidence Interval	[−0.6814 − 0.6207]	[−0.0121 0.0062]	[0.9935 1.0126]	[−0.0485 0.0220]	[0.6038 0.6706]

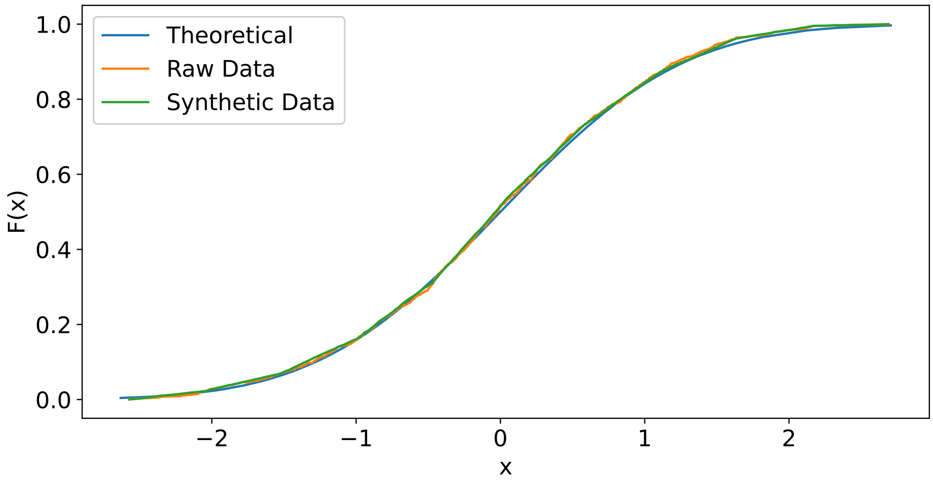


Figure 2. Theoretical distribution and empirical distribution of raw data and synthetic data.

Example 2. *Let us now simulate 300 data points from an exponential distribution with mean 5. These 300 data points will be considered as an observed sample, i.e., raw data. From these raw data, we will generate 1000 synthetic data points. We followed the same steps as in the previous example. Table 3 presents relevant parameters for these data*

From the same raw data, we generate 2000 collections of synthetic data of same size 1000. Table 4 displays the mean value and the 95% confidence interval of values considered in Table 3.

Figure 3 shows the curve of the theoretical distribution, the empirical distribution of raw data and the empirical distribution of synthetic data where a good fit can be observed.

Table 3. Population parameters of Exponential distribution and their point estimates considering the raw data and synthetic data generated from those raw data.

Exponential	P_{25}	Mean	Std	Median	P_{75}
Theoretical Value	1.4384	5	5	3.4657	6.9315
Raw Data	1.9618	5.1615	4.6496	3.8424	7.1929
Synthetic Data	1.7346	5.2074	4.5679	4.0704	7.5514

Table 4. Mean value and confidence intervals for the 2000 synthetic data collections.

Exponential	P_{25}	Mean	Std	Median	P_{75}
Mean Value	1.8482	5.2006	4.5643	3.9670	7.4234
Confidence Interval	[1.7104 1.9759]	[5.1474 5.2500]	[4.5147 4.6143]	[3.8131 4.1276]	[7.1650 7.6706]

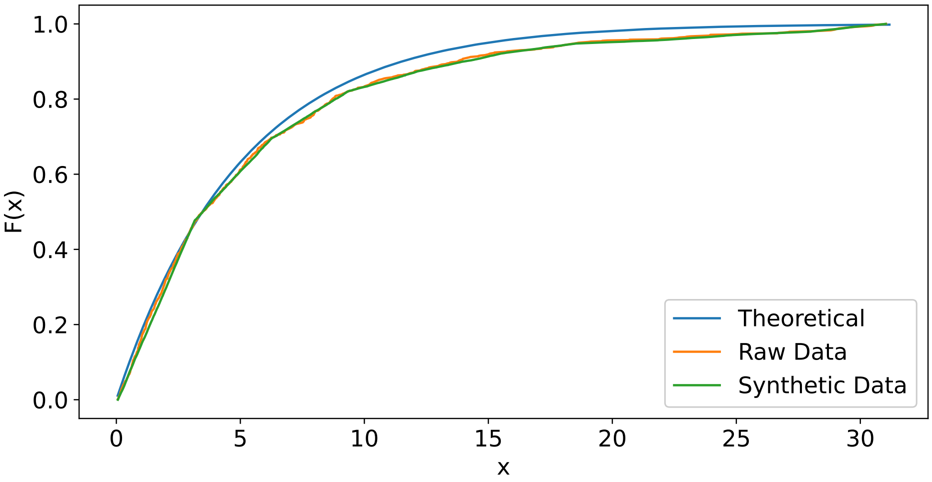


Figure 3. Theoretical distribution and empirical distribution of raw data and synthetic data.

Once the methodology to generate synthetic data from a univariate raw data sample has been designed, then we propose a technique to generate multivariate synthetic data from a multivariate data sample that comes from a random vector $\mathcal{X} = [X_1, \dots, X_p]$ with multivariate distribution F, with F_i being the marginal distribution of the random variable X_i.

Let $\mathbf{X}$ be a $\mathbb{R}^{n \times p}$ a matrix of the real data that comes from the random vector $\mathcal{X}$

$$\mathbf{X} = \begin{pmatrix} X_{11} & \cdot & \cdot & \cdot & X_{1p} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ X_{n1} & \cdot & \cdot & \cdot & X_{np} \end{pmatrix} \tag{3}$$

$\mathbf{X}$ is a raw data multivariate sample; one could think that, for generating synthetic data from $\mathbf{X}$, it is enough to generate, for each variable $X_i, \;\; i = 1, \cdots, p$, univariate synthetic data through the method introduced above, but the procedure is more complex because one must consider the dependence structure among marginal variables X_i. This dependence structure must be studied using a special dependence functions called copulas. We consider Nelsen [35] as the foremost general reference for an in-depth examination of copula theory.

A copula is a multivariate distribution function defined on $[0,1]^p$, where each of the p marginal distributions is a uniform distribution in $[0,1]$. According to Sklar's theorem [17], any multivariate distribution function can be written in terms of the marginal distributions and a copula C, i.e., given a p-dimensional random vector $\mathcal{X} = [X_1, \dots, X_p]$ with a multivariate distribution F, then

$$F(x_1, \dots, x_p) = P(X_1 \leq x_1, \dots, X_p \leq x_p) = C(F_1(x_1), \dots, F_p(x_p)).$$

Sklar's theorem also states that the copula C is unique if the marginals $F_i(x_i)$ are continuous. The copula C contains the information on the dependence structure among the marginal random variables of $\mathcal{X} = [X_1, \dots, X_p]$, and this dependence structure is an important aspect that must be considered when synthetic data are generated from the multivariate sample stated in the matrix of the real data **X** that comes from the random vector $\mathcal{X}$. From Equation (1), $F_i(X_i) \sim$ uniform[0, 1], thus the random vector

$$[U_1, \ldots, U_p] = [F_1(X_1), \ldots, F_p(X_p)] \tag{4}$$

has uniform marginals in $[0, 1]$. Since each of F_i is a non decreasing function, then the random vectors $[U_1, \ldots, U_p]$ and $[X_1, \ldots, X_p] = [F_1^{-1}(U_1), \ldots, F_p^{-1}(U_p)]$ have the same copula because the copula is invariant under non-decreasing transformations of the marginal random variables. Therefore, if there is a procedure to generate, from a known copula C, observations of the random vector $[U_1, \ldots, U_p]$, then a sample from $[X_1, \ldots, X_p]$ can be obtained as $[F_1^{-1}(U_1), \ldots, F_p^{-1}(U_p)]$; here, the marginal distributions also must be known. However, when neither the copula C nor the marginal distributions F_i are known, only a real data multivariate sample as **X** is known, the previous procedure can not be implemented. Therefore, we introduce a new method for the generation of a new multivariate sample just knowing the multivariate sample stated in the matrix of the real data **X**:

Define

$$\hat{U}_{ji} = \frac{1}{n} \sum_{k=1}^{n} I_{\{X_{ki} \leq X_{ji}\}}, \ \ \forall\, i \in \{1, \cdots, p\}, \tag{5}$$

where $I_{\{\cdot\}}$ is the indicator function. Note that $\hat{U}_{ji}$ is the empirical distribution of observed value X_{ji} with respect to the observed sample just of the i-th variable. Therefore, following Equation (4), the following matrix,

$$\hat{U} = \begin{pmatrix} \hat{U}_{11} & \cdot & \cdot & \cdot & \hat{U}_{1p} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \hat{U}_{n1} & \cdot & \cdot & \cdot & \hat{U}_{np}, \end{pmatrix} \tag{6}$$

is the p-dimensional support of the empirical copula estimated from multivariate sample **X**, where the columns of $\hat{U}$ are samples that come from random vector $[U_1, \ldots, U_p]$ considered in Equation (4) and thus those samples come from the same copula C, which is the same copula that has the random vector $\mathcal{X} = [X_1, \ldots, X_p]$, from which comes the raw data multivariate sample stated in matrix **X**. Since the copula C is unknown, the new procedure introduced just considers the p-dimensional support of the empirical copula estimated from multivariate sample **X** as follows.

For the i-th column in **X**, consider a partition of the interval $[X_{[1]i},\ X_{[n]i}]$ as $X_{[1]i} = a_{0i} < a_{1i} < \cdots < a_{t_i i} = X_{[n]i}$. Define B_{si} as:

$$B_{si} = \begin{cases} [a_{(s-1)i},\ a_{si}] & \text{if } s = 1 \\ \\ (a_{(s-1)i},\ a_{si}] & \text{otherwise} \end{cases}$$

$$R(B_{si}) = \frac{1}{n} \sum_{j=1}^{s} \sum_{k=1}^{n} I_{\{X_{ki} \in B_{ji}\}}, \quad \forall\, s \in \{1, \cdots, t_i\}. \tag{7}$$

Note that $R(B_{si})$ can be seen as a natural estimator for $F_i(a_{si})$, and from Proposition 1, the vector $\left[R(B_{s1}), \ldots, R(B_{sp})\right]$ is an unbiased estimator of the vector $\left[F_i(a_{s1}), \ldots, F_i(a_{sp})\right]$.

Let d be a value generated from a discrete uniform distribution in $\{1, \ldots, n\}$. Since the copula C is unknown, we use the the p-dimensional support of the empirical copula by selecting the d-th row of $\hat{U}$. Following the same idea considered in the inverse transform, one must generate a random variable U Uniform in [0, 1]; then, for each $i \in \{1, \ldots, p\}$, calculate $\min\{s \mid R(B_{si}) \geq \hat{U}_{di}\}$ and so a multivariate synthetic piece of data $[\hat{X}_1, \ldots, \hat{X}_p]$, with the same statistics structure as marginals and dependence of those data stated in the matrix of the real data $\mathbf{X}$, can be generated by considering $\hat{X}_i = a_{(s-1)i} + (a_{si} - a_{(s-1)i})U$ for all $i \in \{1, \ldots, p\}$.

Algorithm 1 synthesizes the set of steps formalized above. The described procedure is implemented in the GitHub repository https://github.com/jurest82/SyntheticDataCopulas (accessed on 31 January 2023) using Python3.

Algorithm 1: Synthetic Data Generation Algorithm

```
Input:
$\mathbf{X} \leftarrow \mathbb{R}^{n \times p}$ matrix of the real data
$N \leftarrow$ number of synthetic observations to generate
$T \leftarrow \mathbb{R}^{p}$ selected number of bins of every variable in $\mathbf{X}$
Output:
$\mathbf{Y} \leftarrow \mathbb{R}^{N \times p}$ matrix of the synthetic data
Initialize $\mathbf{U}$ as an array of zeros of size $n \times p$
Initialize $\mathbf{Y}$ as an array of zeros of size $N \times p$
Initialize $\mathbf{D}$ as a list of size $N$ filled with randomly and equiprobably selected
  integers between 1 and n
for $i \leftarrow 1$ to $p$ do
    Generate and store the empirical distribution function for the $i$-th variable in $\mathbf{X}$
    Generate and store the frequency tables with $T[i]$ bins for the $i$-th variable in $\mathbf{X}$
Initialize a counter variable $count$ as zero
for $i \leftarrow 1$ to $p$ do
    for $j \leftarrow 1$ to $n$ do
        $U[j,i] \leftarrow$ the empirical distribution function value for $\mathbf{X}[j,i]$
for $d$ in $D$ do
    Initialize $K$ as an array of zeros of size $1 \times p$.
    for $i \leftarrow 1$ to $p$ do
        Find the corresponding class interval for $\mathbf{U}[d,i]$ in the respective frequency
          table for the $i$-th column
        Generate a uniformly distributed number in the corresponding class
          interval of $\mathbf{U}[d,i]$
        Store the generated number in $K[1,i]$
    Replace row number $count$ in $\mathbf{Y}$ for $K$
    $count \leftarrow count + 1$
```

To understand how the performance of the algorithm scales related to the size of the parameters N, n, p and T, we analyze the complexity of Algorithm 1. First, computing the empirical distribution function along the frequency tables for all variables is $\mathcal{O}(p\ n\ log(n))$ time. In addition, the worst time of evaluating an empirical distribution function is $\mathcal{O}(log(n))$, and repeated to every element in matrix $\mathbf{X}$ is then $\mathcal{O}(p\ n\ log(n))$. Let $t = max_{i \in \{1,\ldots,p\}} T[i]$, and the creation of a datum is bounded by $\mathcal{O}(p\ log(t))$ time. Therefore, the generation of a sample of size N is $\mathcal{O}(N\ p\ log(t))$ time. Thus, the worst time complexity of the **Synthetic Data Generation Algorithm** is $\mathcal{O}(p\ n\ log(n)\ +\ N\ p\ log(t))$ and since t is bounded by n, the expression can be simplified to $\mathcal{O}(max(N,\ n)\ p\ log(n))$.

2.2. Experiments

To illustrate the goodness of the method, several experiments were carried out. The first group of experiments corresponds to generating **SD** from realizations of **RD** following a distribution $\mathcal{F}$. The generated data and the real data are compared using scatter plots since they are presented only as an illustration. To generate the data $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^p$, p = 2, 3, are sampled from $\mathcal{F}$. The patterns with which these real data are obtained are identified as (**a**) cubic function, (**b**) two-dimensions spiral, (**c**) Batman logo, (**d**) petals and (**e**) three-dimensions spiral. These cases were chosen since, in general, they present patterns with complex geometries, which allows us to demonstrate that, even in complex cases, the data generation method works correctly.

The second group of experiments corresponds to the generation of data from the real data set known as *Wine Quality Data Set*, which corresponds to data on the physicochemical properties of red wine and is taken from [38]. For this experiment, the **RD** are compared with the synthetic ones using scatter plots, probability density plots and with the multivariate homogeneity test explained in Section 2.3. This case is presented to show that the method maintains good results even in high-dimensional data sets.

2.3. Homogeneity Test

Liu et al. [36] present a method to determine if two multivariate distributions come from the same population. The method is based on data depth plots of their samples. In this article, we use the above method to show that the samples generated by the data augmentation algorithm follow the same multivariate distribution as the **RD**.

Liu et al. [36] propose the **DD-plot**, which corresponds to a plot of the combined sample depth under the two corresponding empirical distributions. If the distributions are identical, i.e., come from the same population, then the depth plot is a segment of a straight line joining the points $(0,0)$ and $(1,1)$ in $\mathbb{R}^2$. If there is any deviation of said graph from the straight line, then it is a sign that the distributions are not identical.

Formally, let $D(\cdot)$ be an affine invariant depth and let F and G be two distributions on $\mathbb{R}^p$, and $DD(F,G)$ is defined as follows:

$$DD(F,G) = \{(D_F(z), D_G(z)) \text{ for all } z \in \mathbb{R}^p\}$$

If the distributions are unknown, then an empirical version of the **DD-plot** must be used; for example, if F and G are a set of observations $\{X_1, X_2, ..., X_n\}(\equiv \mathbf{X})$, $\{Y_1, Y_2, ..., Y_m\}(\equiv \mathbf{Y})$, respectively, the **DD-plot** is defined as follows:

$$DD(F_n, G_m) = \left\{ \left(D_{F_n}(z_j), D_{G_m}(z_j)\right), z_j \in \{\mathbf{X} \cup \mathbf{Y}\} \right\} \tag{8}$$

Since, in practice, we do not know the multivariate distribution, then the expression given by Equation (8) was used in our experiments. In the particular case of this investigation, the depth measurement is given by a normalization of the metric induced by the p norm:

$$\begin{aligned} D_{F_n}(z_j) &= 1 - \frac{\sum_{i=1}^{n} \left\| X_i - z_j \right\|_p}{\sum_{j=1}^{n+m} \sum_{i=1}^{n} \left\| X_i - z_j \right\|_p}, \forall z_j \in \{\mathbf{X} \cup \mathbf{Y}\} \\ D_{G_m}(z_j) &= 1 - \frac{\sum_{i=1}^{m} \left\| Y_i - z_j \right\|_p}{\sum_{j=1}^{n+m} \sum_{i=1}^{m} \left\| Y_i - z_j \right\|_p}, \forall z_j \in \{\mathbf{X} \cup \mathbf{Y}\}. \end{aligned} \tag{9}$$

In particular, we use $p = 2$ in Equation (9), that is, the Euclidean norm.

To prove by statistical inference that the points $(D_{F_n}(z_j), D_{G_m}(z_j))$ lie on the straight line joining the points $(0,0)$ and $(1,1)$, it is enough to show that the tuples follow a linear relationship and that, if a simple linear regression is fitted, then the confidence interval of the intercept β_0 contains zero, the confidence interval of the slope β_1 contains one, and the coefficient of determination R^2 is very close to one. We follow the approach described to show that the generated data come from the same distribution as the original data; for this, we estimate bootstrap confidence intervals of 95% by the percentile method, as explained in ([37], pp. 27–39).

2.4. Sensitivity Analysis

Section 2.1 discusses an essential parameter of the method known as the number of partitions represented by t. This parameter determines the number of bins or partitions in a frequency table used to generate **SD**. In this section, we explain the simulation analysis carried out to study the effect of t on the quality of the synthetic data.

We run a simulation using the raw data from the first scatterplot in Figure 1. We generate new data with fixed t and the same value for both dimensions of the plot. We repeated this process for six different values of $t = 5, 10, 20, 30, 50$, and 100. **SD** was visually compared using scatterplots.

3. Results and Analysis

In this section, the obtained results from the experiments explained in Section 2.2 are analyzed. Figure 4 presents five pairs of scatter plots of the following patterns: (**a**) cubic function, (**b**) two-dimensions spiral, (**c**) Batman logo, (**d**) petals, and (**e**) three-dimensions spiral. Blue dots represent real data, while red dots correspond to synthetically generated data. As can be seen, the plots of the **SD** are very similar to the plots of the real observations. No matter what geometry needs to be replicated, the data augmentation method can learn the underlying dependency relationships and probability distributions. This first group of experiments allows us to demonstrate graphically and simply the goodness of the synthesis method.

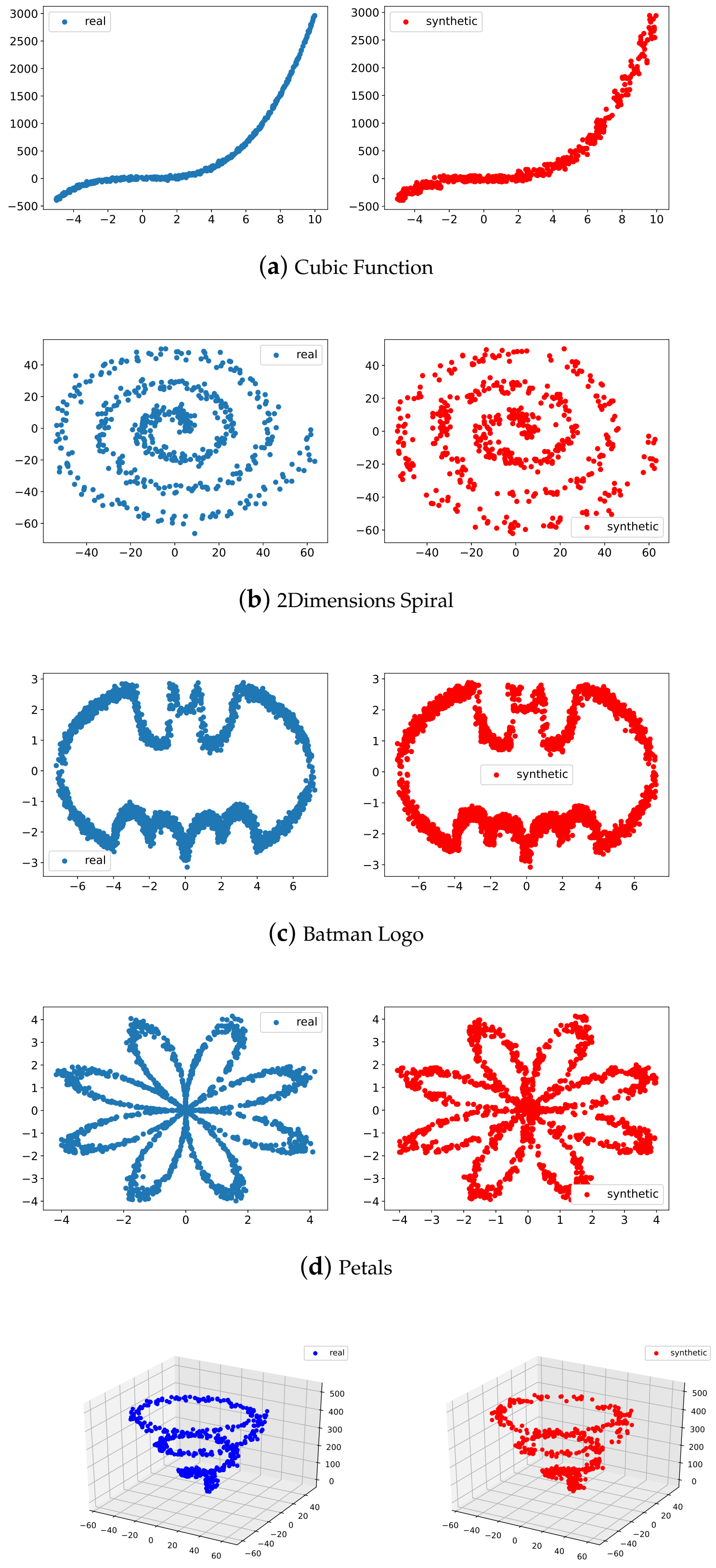
Regarding the wine quality data obtained from [38], we must observe the respective scatter and probability density plots depicted in Figure 5.

Figure 5 shows that the geometric shape of the probability distributions are identical, that is, the **SD** generation method respects the marginal distributions of each variable. Moreover, when analyzing the scatter plots, it is observed that the bivariate dependency relationships are also respected by the proposed method since the geometric shape and the scale are identical to those of the **RD**. In practice, these facts indicate that synthetic data are similar to real data in univariate and bivariate parameters. Univariate parameters, such as mean, median, standard deviation, interquartile range, kurtosis, and skewness, are identical in **SD** and **RD**. Bivariate parameters, such as covariance, Pearson's correlation, and Spearman's correlation, are identical for both data sets. This indicates that the **SD** accurately captures the relationships and patterns present in the **RD**.

However, as explained before, to conclude that the joint distribution of the real and synthetic data are equivalent, it is not enough to look at the marginal distributions and the bivariate dependency relationships. A homogeneity test needs to be used to conclude whether two multivariate data sets have the same joint distribution.

Figure 6 corresponds to the **DD-plot** explained in Section 2.3, where F_n corresponds to the original data and G_m to the generated samples. It should be noted that the depth values form a straight line that, fitting a simple linear regression model, yields a confidence interval for R^2 of $[0.9922, 0.9999]$, a confidence interval for intercept β_0 of $[-0.0957, 0.0618]$ and a confidence interval for the slope β_1 of $[0.9069, 1.1087]$. Therefore, the homogeneity test confirms that the generated samples come from the same multivariate distribution as the original data, since the line they form is a segment of the straight line that joins the points $(0, 0)$ and $(1, 1)$ in $\mathbb{R}^2$ because R^2 is very close to one, and it cannot be rejected that β_0 is zero and that β_1 is one with a confidence level of 95%.

In Section 2, we mentioned that the number of bins used in the frequency tables can significantly affect the result of Algorithm 1. To illustrate this, Figure 7 shows six data sets generated from the raw data contained in the first scatterplot of Figure 1. Each of these data sets was created using a different number of partitions.



(**a**) Cubic Function

(**b**) 2Dimensions Spiral

(**c**) Batman Logo

(**d**) Petals

(**e**) 3Dimensions Spiral

Figure 4. Scatter plots of the first group of experiments.

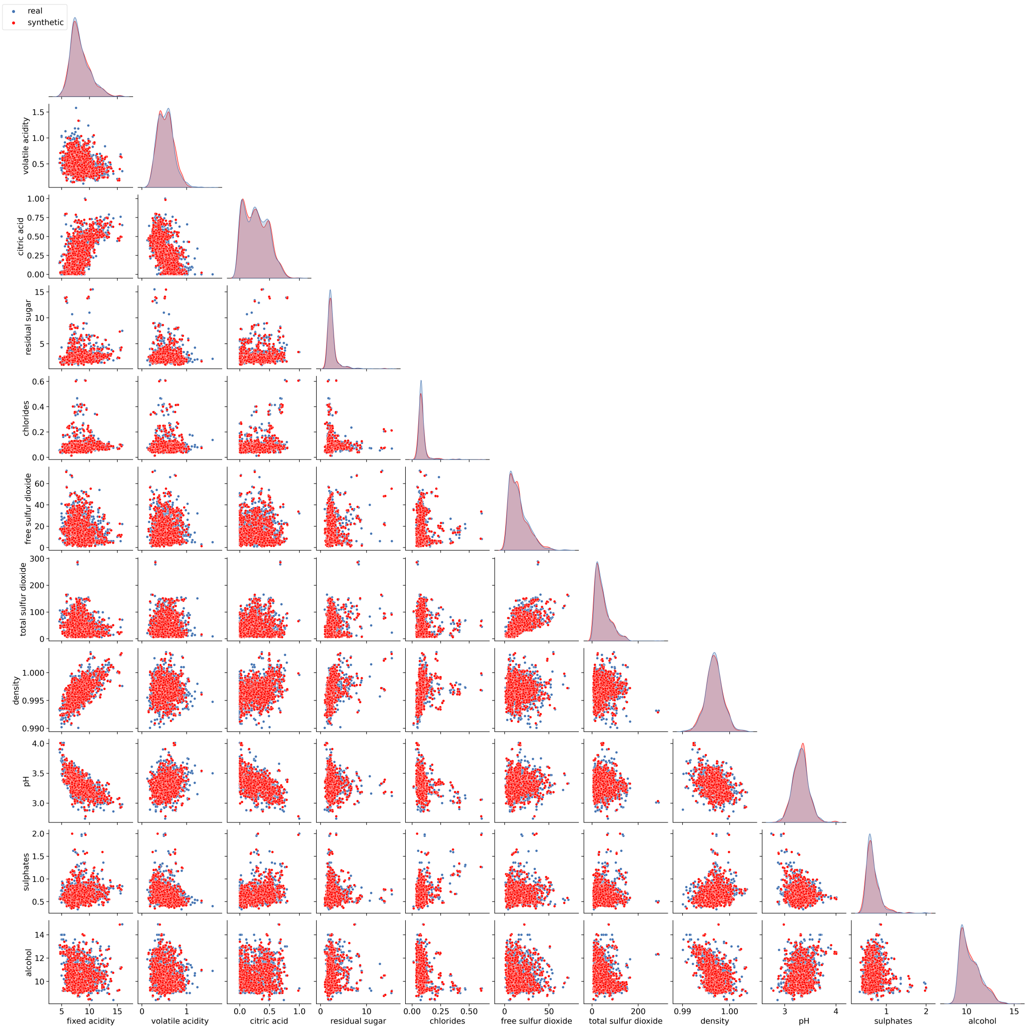


Figure 5. Scatter plots and density plots of the second group of experiments.

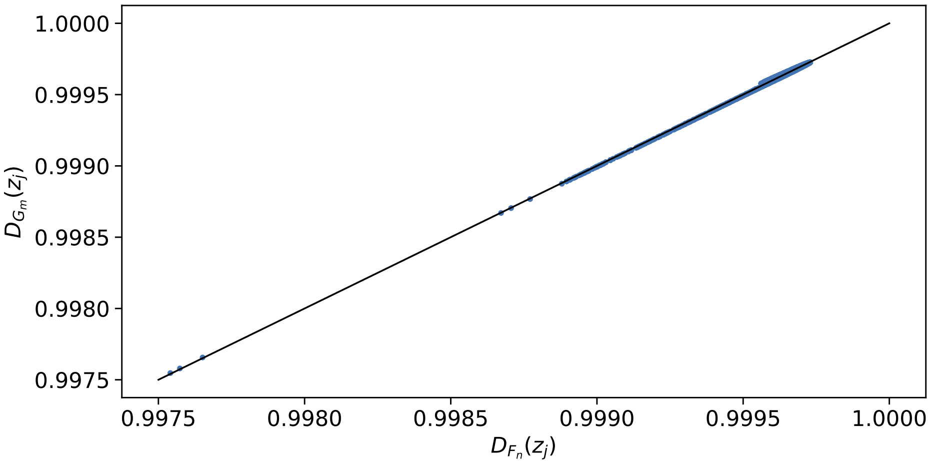


Figure 6. DD-plot of synthetic data vs. real data.

As shown in Figure 7, the number of bins used has a noticeable effect on the similarity between synthetic and real data sets. Specifically, we find that increasing the number of bins results in **SD**, which more closely reflects the dependency structure of the raw data. Conversely, using a low number of bins fails to capture this structure accurately. However, a very high number of partitions can produce synthetic data with less variability.

The question of how to select the optimal number of partitions for a frequency table, histogram, or density estimate is outside the scope of this research. For more in-depth analysis on this topic, interested readers may refer to works such as Hollander et al. ([39], pp. 609–628), Silverman [40], and Wasserman ([37], pp. 125–142), among others.

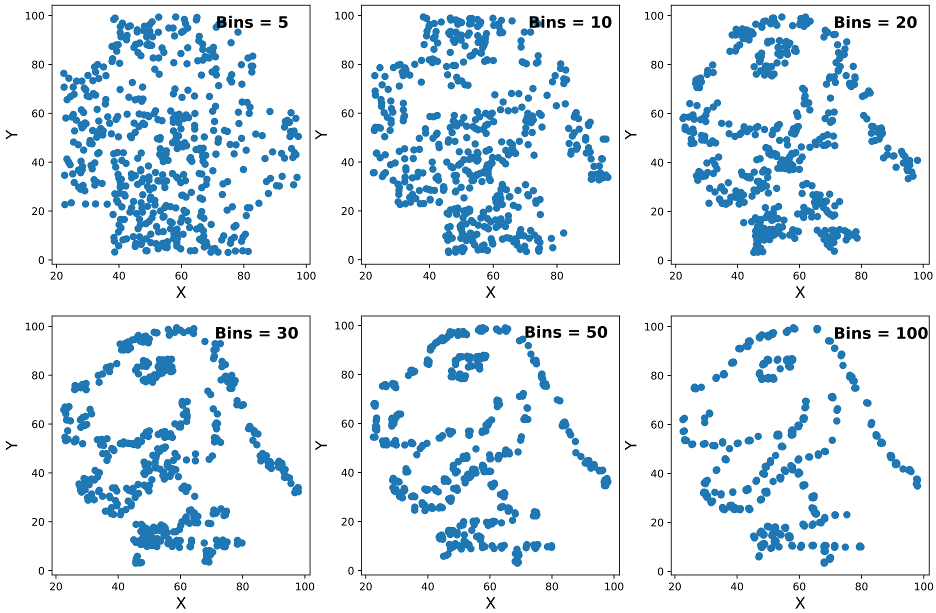


Figure 7. Evolution of the synthetic data for different numbers of partitions or bins.

A notable result of the sensitivity analysis is that the six data sets shown in Figure 7 are marginally equal to each other. To illustrate this point, we present Table 5, which contains summary statistics for the cases where $t = 5$ and $t = 100$. While these statistics are fairly similar between partitions, we note that the method does not capture the multivariate dependency structure effectively when $t = 5$, but it does when $t = 100$. Therefore, we conclude that increasing the value of t can improve the performance of the method.

Table 5. Summary statistics for selected plots of Figure 7.

	Variable	Mean	Std	P_{25}	P_{50}	P_{75}
bins = 5	X	54.9	17.9	40.9	53.9	66.8
	Y	46.3	26.9	24.3	44.7	65.8
bins = 100	X	54.9	16.6	45.5	52.9	64.5
	Y	46.6	28.0	23.5	41.5	71.3

4. Conclusions

This article presents a novel nonparametric method for generating synthetic data using copulas. Said method is easy to use, implement and interpret, but more importantly, synthetic data generated by the method maintain identical marginal and joint distributions and the same dependency relationships as real data. Preserving the statistical properties of raw data are a fundamental property of the algorithm that enhances confidence in using synthetic data for practical applications. This is useful in realistic scenarios such as class balancing in machine learning algorithms, protecting sensitive information in financial and healthcare domains, and more. The aforementioned fact was illustrated with an example in which a multivariate homogeneity test was used, something that is not usually found in the scientific literature on data augmentation. The proposed method is promising, and it will be of interest for future research to use it to increase the performance of machine learning models and adapt it to perform multivariate imputation, among others. Furthermore, determining the optimal way to select the unique hyperparameter of the model, i.e., the number of bins, is relevant for forthcoming studies.

The proposed algorithm has two crucial limitations. First, it only works with tabular data, as it has not been extended to unstructured data such as text, images or audio, which limits its applicability to data augmentation problems in deep learning applications. Expanding the algorithm's compatibility with unstructured data will be a crucial advance in future research. Secondly, the formulation does not account for categorical variables. As a result, the method cannot generate data from a tabular dataset that includes this variable type.

Author Contributions: Conceptualization, H.L.; methodology, H.L., J.C.R., J.P.R., P.O. and O.A.B.; software, H.L., J.P.R. and P.O.; validation, H.L., J.C.R., J.P.R., P.O. and O.A.B.; formal analysis, H.L., J.C.R., J.P.R., P.O. and O.A.B.; investigation, H.L., J.C.R., J.P.R., P.O. and O.A.B.; resources, H.L. and J.C.R.; data curation, J.P.R., P.O. and O.A.B.; writing—original draft preparation, H.L. and J.P.R.; writing—review and editing, H.L., J.C.R., J.P.R., P.O. and O.A.B.; visualization, J.P.R., P.O. and O.A.B.; supervision, H.L. and J.C.R.; project administration, J.C.R.; funding acquisition, H.L. and J.C.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the call 852-2019 of the Ministry of Science, Technology and Innovation of the Republic of Colombia (MinCiencias), which allowed the development of the project with code 1216-852-72082 called "Descriptive and predictive analysis of the cement and concrete production process".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work has been carried out within the framework of the project "Descriptive and predictive analysis of the cement and concrete production process" that has been financed by the Science and Technology program with the Ministry of Science, Technology and Innovation (MinCiencias) of Colombia. The authors also want to thank Daniel Duque, Juan Tobón and Ana Gómez from the company Cementos Argos, and John Fernando Vargas, Leonardo Betancur and Ana Isabel Oviedo from Universidad Pontificia Bolivariana who are part of the research team of the research project and have contributed to the understanding of cement and concrete processes.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

DD-plot	Data-depth plot
GAN	Generative adversarial networks
ML	Machine learning
RD	Real data
SD	Synthetic data
SMOTE	Synthetic minority oversampling technique

References

1. Liang, Y.; Nobakht, B.; Lindsay, G. The application of synthetic data generation and data-driven modelling in the development of a fraud detection system for fuel bunkering. *Meas. Sens.* **2021**, *18*, 100225. [CrossRef]
2. Dilmegani, C. What is Synthetic Data? What Are Its Use Cases & Benefits? 2023. Available online: https://research.aimultiple.com/synthetic-data/ (accessed on 1 January 2023).
3. Rankin, D.; Black, M.; Bond, R.; Wallace, J.; Mulvenna, M.; Epelde, G. Reliability of Supervised Machine Learning Using Synthetic Data in Health Care: Model to Preserve Privacy for Data Sharing. *JMIR Med. Inform.* **2020**, *8*, e18910. [CrossRef]
4. Yale, A.; Dash, S.; Dutta, R.; Guyon, I.; Pavao, A.; Bennett, K.P. Generation and evaluation of privacy preserving synthetic health data. *Neurocomputing* **2020**, *416*, 244–255. [CrossRef]

5. Yoon, J.; Drumright, L.N.; van der Schaar, M. Anonymization Through Data Synthesis Using Generative Adversarial Networks (ADS-GAN). *IEEE J. Biomed. Health Inform.* **2020**, *24*, 2378–2388. [CrossRef]
6. Douzas, G.; Bacao, F. Effective data generation for imbalanced learning using conditional generative adversarial networks. *Expert Syst. Appl.* **2018**, *91*, 464–471. [CrossRef]
7. Ahmed, J.; Green II, R.C. Predicting severely imbalanced data disk drive failures with machine learning models. *Mach. Learn. Appl.* **2022**, *9*, 100361. [CrossRef]
8. Moreno-Barea, F.J.; Franco, L.; Elizondo, D.; Grootveld, M. Application of data augmentation techniques towards metabolomics. *Comput. Biol. Med.* **2022**, *148*, 105916. [CrossRef] [PubMed]
9. Temraz, M.; Keane, M.T. Solving the class imbalance problem using a counterfactual method for data augmentation. *Mach. Learn. Appl.* **2022**, *9*, 100375. [CrossRef]
10. Lashgari, E.; Liang, D.; Maoz, U. Data augmentation for deep-learning-based electroencephalography. *J. Neurosci. Methods* **2020**, *346*, 108885. [CrossRef]
11. Porcu, S.; Floris, A.; Atzori, L. Evaluation of Data Augmentation Techniques for Facial Expression Recognition Systems. *Electronics* **2020**, *9*, 1892. [CrossRef]
12. Andreini, P.; Ciano, G.; Bonechi, S.; Graziani, C.; Lachi, V.; Mecocci, A.; Sodi, A.; Scarselli, F.; Bianchini, M. A Two-Stage GAN for High-Resolution Retinal Image Generation and Segmentation. *Electronics* **2021**, *11*, 60. [CrossRef]
13. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
14. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Netsworks: An overview *IEEE Signal Process. Mag.* **2018**, *35*, 53–65.
15. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
16. Poudevigne-Durance, T.; Jones, O.D.; Qin, Y. MaWGAN: A Generative Adversarial Network to Create Synthetic Data from Datasets with Missing Data. *Electronics* **2022**, *11*, 837. [CrossRef]
17. Sklar, A. Fonctions de Répartition à n Dimensions et Leurs Marges. *Publ. L'Institut Stat. L'UniversitÉ Paris* **1959**, *8*, 229–231.
18. Patki, N.; Wedge, R.; Veeramachaneni, K. The Synthetic Data Vault. In Proceedings of the 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Montreal, QC, Canada, 17–19 October 2016; pp. 399–410. [CrossRef]
19. Nejad, M.M.; Erdogan, S.; Cirillo, C. A statistical approach to small area synthetic population generation as a basis for carless evacuation planning. *J. Transp. Geogr.* **2021**, *90*, 102902. [CrossRef]
20. Li, Z.; Zhao, Y.; Fu, J. SynC: A Copula based Framework for Generating Synthetic Data from Aggregated Sources. In Proceedings of the 2020 International Conference on Data Mining Workshops (ICDMW), Sorrento, Italy, 17–20 November 2020, Volume 2020; pp. 571–578. [CrossRef]
21. Benali, F.; Bodénès, D.; Labroche, N.; de Runz, C. MTCopula: Synthetic Complex Data Generation Using Copul. In Proceedings of the 23rd International Workshop on Design, Optimization, Languages and Analytical Processing of Big Data (DOLAP), Nicosia, Cyprus, 23 March 2021; Volume 2840, pp. 51–60.
22. Endres, M.; Mannarapotta Venugopal, A.; Tran, T.S. Synthetic Data Generation: A Comparative Study. In Proceedings of the International Database Engineered Applications Symposium, Budapest Hungary, 22–24 August 2022; ACM: New York, NY, USA, 2022; pp. 94–102. [CrossRef]
23. Reiter, J.P. Using CART to generate partially synthetic, public use microdata. *J. Off. Stat.* **2005**, *21*, 441–462.
24. Ping, H.; Stoyanovich, J.; Howe, B. DataSynthesizer: Privacy-Preserving Synthetic Datasets. In Proceedings of the 29th International Conference on Scientific and Statistical Database Management, Chicago, IL, USA, 27–29 June 2017; ACM: New York, NY, USA, 2017; pp. 1–5. [CrossRef]
25. Wang, Q.; Gao, J.; Lin, W.; Yuan, Y. Pixel-Wise Crowd Understanding via Synthetic Data. *Int. J. Comput. Vis.* **2021**, *129*, 225–245. [CrossRef]
26. Boikov, A.; Payor, V.; Savelev, R.; Kolesnikov, A. Synthetic Data Generation for Steel Defect Detection and Classification Using Deep Learning. *Symmetry* **2021**, *13*, 1176. [CrossRef]
27. Shamsolmoali, P.; Zareapoor, M.; Zhou, H.; Wang, R.; Yang, J. Road Segmentation for Remote Sensing Images Using Adversarial Spatial Pyramid Networks. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 4673–4688. [CrossRef]
28. Farajzadeh-Zanjani, M.; Hallaji, E.; Razavi-Far, R.; Saif, M.; Parvania, M. Adversarial Semi-Supervised Learning for Diagnosing Faults and Attacks in Power Grids. *IEEE Trans. Smart Grid* **2021**, *12*, 3468–3478. [CrossRef]
29. Hernandez, M.; Epelde, G.; Beristain, A.; Álvarez, R.; Molina, C.; Larrea, X.; Alberdi, A.; Timoleon, M.; Bamidis, P.; Konstantinidis, E. Incorporation of Synthetic Data Generation Techniques within a Controlled Data Processing Workflow in the Health and Wellbeing Domain. *Electronics* **2022**, *11*, 812. [CrossRef]
30. Gonzalez-Abril, L.; Angulo, C.; Ortega, J.A.; Lopez-Guerra, J.L. Statistical Validation of Synthetic Data for Lung Cancer Patients Generated by Using Generative Adversarial Networks. *Electronics* **2022**, *11*, 3277. [CrossRef]
31. Dankar, F.K.; Ibrahim, M.K.; Ismail, L. A Multi-Dimensional Evaluation of Synthetic Data Generators. *IEEE Access* **2022**, *10*, 11147–11158. [CrossRef]
32. Hernadez, M.; Epelde, G.; Alberdi, A.; Cilla, R.; Rankin, D. Synthetic Tabular Data Evaluation in the Health Domain Covering Resemblance, Utility, and Privacy Dimensions. *Methods Inf. Med.* **2023**. [CrossRef]

33. Matejka, J.; Fitzmaurice, G. Same Stats, Different Graphs. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, CO, USA, 6–11 May 2017; ACM: New York, NY, USA, 2017; Volume 2017; pp. 1290–1294. [CrossRef]
34. Matejka, J.; Fitzmaurice, G. Same Stats, Different Graphs: Generating Datasets with Varied Appearance and Identical Statistics through Simulated Annealing. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, CO, USA, 6–11 May 2017.
35. Nelsen, R.B. *An Introduction to Copulas*; Springer Series in Statistics; Springer: New York, NY, USA, 2006. [CrossRef]
36. Liu, R.Y.; Parelius, J.M.; Singh, K. Multivariate analysis by data depth: descriptive statistics, graphics and inference, (with discussion and a rejoinder by Liu and Singh). *Ann. Stat.* **1999**, *27*, 783–858. [CrossRef]
37. Wasserman, L. *All of Nonparametric Statistics*; Springer Texts in Statistics, Springer New York: New York, NY, USA, 2006. [CrossRef]
38. Cortez, P.; Cerdeira, A.; Almeida, F.; Matos, T.; Reis, J. Modeling wine preferences by data mining from physicochemical properties. *Decis. Support Syst.* **2009**, *47*, 547–553. [CrossRef]
39. Hollander, M.; Wolfe, D.A.; Chicken, E. Density Estimation. In *Nonparametric Statistical Methods*; John Wiley & Sons: New York, NY, USA, 2015; pp. 609–628. [CrossRef]
40. Silverman, B. *Density Estimation for Statistics and Data Analysis*; Routledge: New York, NY, USA, 2017. https://www.taylorfrancis.com/books/mono/10.1201/9781315140919/density-estimation-statistics-data-analysis-bernard-silverman (accessed on 1 January 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.