Testing the Performance of synthesizer: An R Package for Synthesizing Data Through Inverse Transform Sampling with Rank Matching

Jacobs, Mishca

Testing the Performance of `synthesizer`

An R Package for Synthesizing Data Through Inverse Transform Sampling with Rank Matching

Mishca Jacobs

First Thesis Advisor: Dr. Mark van der Loo (CBS & Leiden University)
Second Thesis Advisor: Dr. Sanne Willems (Leiden University)

Defended on 29 January, 2025

MASTER THESIS

STATISTICS AND DATA SCIENCE
UNIVERSITEIT LEIDEN

Abstract

Synthetic data are data that have been generated using some model or algorithm to mimic real data. Synthetic data serves as an alternative to real data, often with the aim of preserving the privacy of the entities represented in the real data. Many methods exist for synthetic data generation, with deep learning models emerging as a particularly popular tool. However, synthesizing economic and financial datasets presents unique challenges, in that they often have many missing values, are zero-inflated and tend to have logical and mathematical restrictions (e.g. $X+Y=Z$). Therefore, CBS has designed a non-parametric method to synthesize datasets of this nature that preserves these properties. The method designed implements a simple two-step procedure that uses inverse transform sampling (ITS) for numeric variables and sampling with replacement for categorical or binary variables, followed by a rank-matching procedure. The method is easily implemented in R using the `synthesizer` package.

This thesis examines the literature on synthetic data and implements five key metrics to evaluate the data synthesized using the `synthesizer` method. The methods implemented are the propensity mean squared error (pMSE), comparisons of the percentages of zero values and the percentages of missing values in the original and synthetic data, comparisons of F_1 scores using the train synthetic test real (TSTR) approach, and ratios of the sum of variables to their total. Results for all datasets simulated using `synthesizer` are compared to those simulated using the non-parametric `synthpop` method in R. Non-parametric `synthpop` is a synthetic data generation (SDG) method that has outscored other SDG methods in previous studies.

The methods are implemented on three data types: multivariate normal simulated data, zero-inflated log-normal simulated data, and real data. The simulation studies are designed to consider varied levels of separation between groups in the data, differing strengths of correlations among variables, and various ratios of records to variables.

The results show that `synthesizer` is good at replicating the properties of economic datasets, i.e. missing values, zero-inflation and mathematical restrictions. However, `synthpop` outperforms `synthesizer` in terms of distributional similarity to the original data in settings there are two distinct groups in the data and when variables are highly correlated. While `synthesizer` shows promise as a SDG method, further research is needed.

Contents

Chapter 1

Introduction

The increased availability of and access to data in today's data-driven society has led to increasing concerns regarding privacy and confidentiality issues. This has, in part, resulted in increased apprehension regarding the trade-off between the benefits of extensive data access and the possible harms that result from the misuse of poorly protected data [1]. As such, National Statistics Institutes (NSIs) globally have grappled with the balance between releasing sufficient information to users and safeguarding against disclosing specific record details within micro-data sets (such as those tied to individuals). This challenge has intensified in recent years, with mounting demands for more detailed and expedited data releases [2].

As the National Statistics Institute (NSI) of the Netherlands, Statistics Netherlands (CBS) serves as the sole authority of official statistical data in the country. In this capacity, CBS collects data from citizens and businesses via direct observations, or by accessing government-held administrative data. This grants CBS a substantial repository of individual-level data, that is of interest to researchers beyond its standard statistical program. This data includes datasets featuring economic and financial information.

However, the Statistics Netherlands Acts prohibits CBS from disclosing any information that may be identifiable at an individual level [3]. Section 37 of the Statistics Netherlands Act explicitly states that data "*shall only be published in such a way that no recognisable data can be derived from them about an individual person, household, company or institution, unless, in the case of data relating to a company or institution, there are good reasons to assume that the company or institution concerned will not have any objections to the publication*" [4].

Traditionally, NSIs such as CBS have mitigated these risks by publicly releasing only tabulated aggregates of the underlying micro-datasets or by granting certain users access to specific micro-data in secure environments. However, these methods have limitations, and may not always satisfy users' information needs. Further, in cases where access is granted, stringent and time-consuming measures are required to manage it.

Synthetic data refers to data that have been artificially generated by some model aimed at mimicking real data. As such, synthetic data, when used responsibly, facilitates the use of datasets where the privacy of the data needs to be protected, and in instances of incomplete, sparse or biased data. Oftentimes, synthetic data is created to replace real data since synthetic data protects sensitive information present in real datasets by providing realistic alternatives that meet analytical needs while, significantly reducing the risk of disclosing details from the original data [2]. However, synthetic data also has applications in many other realms, for example, in the verification and validation of machine-learning pipelines [5]. The application of synthetic data thus emerges as an alternative for NSIs in managing data releases [2]. Therefore, CBS is interested in generating synthetic data that may be used by researchers and possibly individuals in the wider public.

Various approaches exist for synthetic data generation (SDG). However, synthesizing datasets containing economic and financial information about enterprises presents unique challenges. These datasets often feature variables with zero-inflated distributions that are highly skewed individually, yet exhibit strong linear correlations. This zero inflation arises because questionnaires sent to businesses are often extensive and may contain many categories that do not apply to various entities. These categories are then assigned zero values, leading to zero-inflated datasets. Additionally, these datasets are created based on surveys completed by individual entities and may also be poorly completed at times. Therefore, these datasets also often have many missing values. Moreover, these datasets must adhere to logical and mathematical rules governing the allowable combinations of variable values. For example, a dataset may contain the variables, Cost (C), Profit (P) and Turnover (T), where the following relationship should hold for these variables: $C + P = T$. Traditional methods have proven inadequate for synthesizing datasets of such a complex nature.

Statistics Netherlands (CBS) has developed a non-parametric method for synthesizing economic and financial data to address these challenges. The method was originally used to generate data for testing imputation methods [6] and has since been adapted for synthetic data generation. The proposed method can be implemented directly in R using the newly created `synthesizer` package [7]. The `synthesizer` package combines inverse transform sampling from the empirical quantile functions for each numeric variable and sampling with replacement for each binary or categorical variable with copying the rank order structure from the original dataset.

While `synthesizer` has proven useful for creating datasets to test new methodologies, such as imputation [6] and preliminary analyses of `synthesizer` have shown promising results in the context of synthesizing data with a relatively small number of variables, the method has not yet been extensively tested. Thus, further research is necessary, which is the aim of this project.

In order to effectively evaluate the `synthesizer` package approach, the primary aims of this thesis project are three-fold:

RQ 1 To establish a framework to assess the quality of synthetic data by defining and selecting a comprehensive set of robust metrics that measure the quality of synthetic data.

RQ 2 To evaluate the performance of the `synthesizer` package against the framework of quality measures. Specifically, this thesis project seeks to assess the package's applicability across different use cases, identify the circumstances under which it performs best, determine the conditions under which it may fail, and assess the quality of the synthetic data it produces.

RQ 3 To compare the `synthesizer` package with the established synthetic data generation (SDG) method `synthpop` in R. Non-parametric `synthpop` is well-known SDG method and has outperformed other SDG packages in various analyses [8], making it a suitable benchmark for comparison with `synthesizer`.

The remainder of this thesis report is structured as follows. Background information about synthetic data generation methods and the `synthesizer` method is provided in chapter 2. The results of the first aim of this thesis: the proposed framework for assessing the quality of synthetic data are described in chapter 3. An overview of the methods implemented to evaluate the `synthesizer` package is given in chapter 4. The results pertaining to the performance of the `synthesizer` package against the defined quality framework are presented in chapter 5. Finally, chapter 6 discusses the results of the previous chapter and gives recommendations for future studies.

Chapter 2

Synthetic Data Generation (SDG) Methods

Synthetic data refers to data that have been artificially generated data using some mathematical model or algorithm [5]. Models used for synthetic data generation are predominantly trained to replicate the characteristics and structure of the original data, such that the synthetic data retains similar statistical properties to the original data [9]. Although synthetic data has gained popularity in recent years, the concept of releasing synthetic data as opposed to real data has been around for some time [1]. In his 1993 discussion on statistical disclosure limitation, Donald B. Rubin first proposed the release of synthetic micro-data created using multiple imputations [10]. Since then, the role of synthetic data has grown across various applications. For example, synthetic data can be used in a variety of different fields, from the development of artificial intelligence (AI) models, to software demonstrations [9].

To support these varied applications, synthetic data is typically classified into two main types: fully synthetic and partially synthetic. In a fully synthetic dataset, all the variables in the data are synthesized. Contrarily, in a partially synthetic dataset, only variables that are confidential and carry a high risk of disclosure are synthesized. For example, datasets where some of the information is already available to the public from other databases or where details are available from public documents such as incorporation statements of accounts could be thought to have a high disclosure risk as variables in the data could be linked to this publicly available information [11]. The variables that are not publicly available would thus need to be synthesized, while the other variables would remain unchanged. The `synthesizer` synthetic data generation (SDG) package evaluated in this thesis project synthesizes fully synthetic data.

2.1 Types of Synthetic Data Generation (SDG) Methods

Given the growing popularity and use of synthetic data, numerous approaches have been developed for generating synthetic data. As a result, many different approaches exist for synthesizing data across various settings today.

Artificial Intelligence (AI) is at the forefront of synthetic data generation (SDG) methods. Many SDG methods use machine learning or deep learning algorithms to synthesize data. Machine learning algorithms generate predictions and recommendations by learning patterns from the data instead of relying on explicit programming instructions. As a result, machine learning algorithms are flexible and adaptable to new data, allowing them to improve as they process new or additional data. Deep learning is a more sophisticated form of machine learning that is particularly effective at processing a broader range of data types, including text and unstructured data like images. Deep learning methods rely on the use of neural networks. These neural networks are inspired by the way neurons interact in the human brain, ingesting and processing data through multiple layers of neurons. These neuron layers are able to recognize increasingly complex features of the data. Deep learning can, therefore, yield more precise results than traditional machine learning. However, AI models that rely on neural networks require large amounts of data to train. [12]

Machine learning models such as classification and regression trees (CART) are common methods for synthesizing data. Deep learning methods such as Variational AutoEncoders [13] (VAEs), Generative adversarial networks [14] (GANs), auto-regressive models and Synthetic Minority Oversampling Technique (SMOTE) methods have also proven to be promising SDG techniques [2].

While these SDG methods have been useful in various applications, many of them are rather complex to implement and may require intensive computational power. Further, these methods may not accurately reflect the relationships or dependencies present in economic and financial data. In contrast, the `synthesizer` package [7] is straightforward to implement and is specifically designed to accommodate the structural complexities of economic datasets. Moreover, contrarily to deep learning methods the `synthesizer` package requires only moderate amounts of data. Typical economic datasets result from surveys across various economic sectors, thus the number of records in these datasets typically varies from a few hundred to a few thousand.

2.2 Proposed Synthetic Data Generation (SDG) Method: `synthesizer`

The `synthesizer` package [7] was developed at Statistics Netherlands (CBS). The method was first used to generate data for testing imputation methods [6], and has since been adapted for synthetic data generation due to a lack of efficient methods for synthesizing financial and economic datasets. These datasets often present unique challenges, as they feature variables with zero-inflated distributions that are highly skewed individually yet exhibit strong linear correlations. Additionally, these datasets contain logical and mathematical rules governing the allowable combinations of variable values. This method is therefore designed to conserve several key data properties typical of datasets containing economic and financial information, namely right-skewed distributions, zero-inflated distributions, linear inequalities, linear restrictions and linear correlations [6].

2.2.1 The `synthesizer` Algorithm

The `synthesizer` package implements a non-parametric approach to generating fully synthetic data aimed at overcoming the aforementioned challenges. The method relies on inverse transform sampling for numeric variables and sampling with replacement for binary or categorical variables, followed by a rank-matching procedure. Inverse transform sampling (ITS) is a sampling technique often used for simulation and as a random number generator. ITS can be used in any setting where the inverse of the cumulative density function (CDF) of a distribution can be calculated [15]. The method implemented in the `synthesizer` packaged is outlined in Algorithm 1.

Algorithm 1 `synthesizer`Algorithm for Synthetic Data Generation

Input: Original dataset X with variables $X_1, X_2, \ldots, X_p$ and n records

Output: Synthetic dataset with m records

Step 1: For each variable X_j:

1.1. If X_j is numeric:

1.1.1. Estimate the Empirical Cumulative Distribution Function (eCDF):

Sort the data points of X_j (denoted $x_1, x_2, \ldots, x_n$), where missing values are sorted to the end. Compute the eCDF $F_{\text{eCDF}}(x_i) = \frac{i}{(n+1)}$ for each sorted value x_i, where i is the index of x_i after sorting and n is the number of records. Use the observed minimum and maximum values of X_j as the minimum and maximum of the new synthesized variable.

Linearly interpolate between the sorted values x_i based on their eCDF values to obtain the inverse eCDF function $F_j^{-1}(p)$, where $p \in \left(\frac{1}{(n+1)}, \frac{n}{(n+1)}\right)$.

1.1.2. Generate Synthetic Samples:

For each synthetic sample, generate a random value $U \sim \text{Uniform}\left(\frac{1}{(n+1)}, \frac{n}{(n+1)}\right)$.Use the inverse eCDF to determine the synthetic data point: $x' = F_{X_j}^{-1}(U)$. Repeat this process n times to generate n synthetic data points.

1.2. Else If X_j is categorical or logical:

Sample from n values from X_j with replacement to get synthetic sample x'.

Step 2: For each synthetic variable X'_j, apply rank matching:

Rearrange each synthetic variable to ensure rank order matches the original dataset's rank order. For categorical and logical variables, the ranks are determined by their alphabetic order (lexicographical ordering).

Note: To synthesize datasets of different sizes (i.e. $m \neq n$)

- **If** $m > n$: Create $\lceil m/n \rceil$ synthetic datasets, each of size n. Combine these datasets and sample m records uniformly without replacement from the combined synthesized data.
- **If** $m < n$**:** Sample m records uniformly without replacement from the synthesized dataset of size n

Figure 1 and Figure 2 visualise step 1 (for the case of numeric variables) and step 2 of the `synthesizer` algorithm, respectively. Note: these figures were created for demonstration purposes only and were not created using real data.

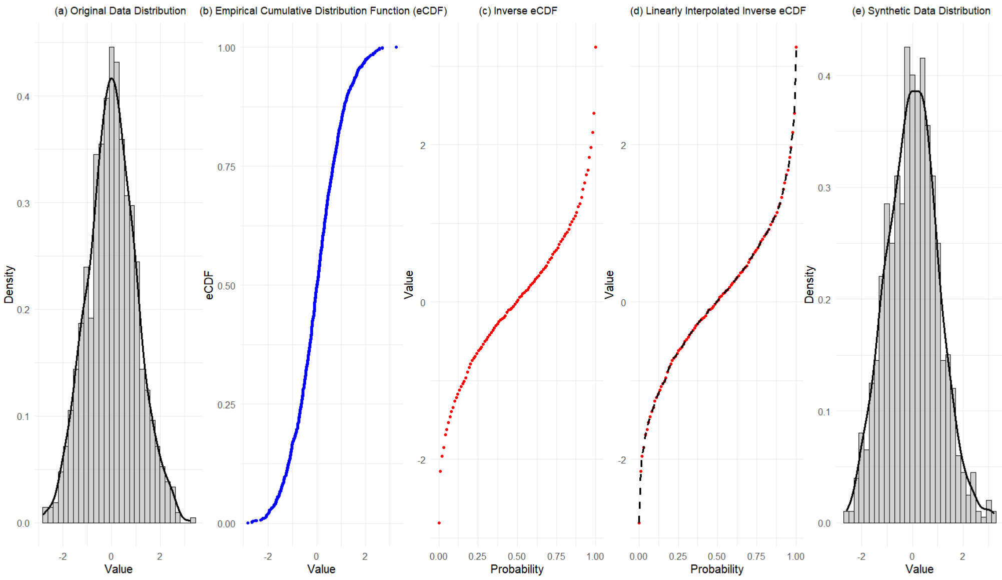


Figure 1: A series of plots demonstrating the process of inverse transform sampling (ITS). ITS is the first step in the `synthesizer` algorithm and is implemented for each numeric variable.

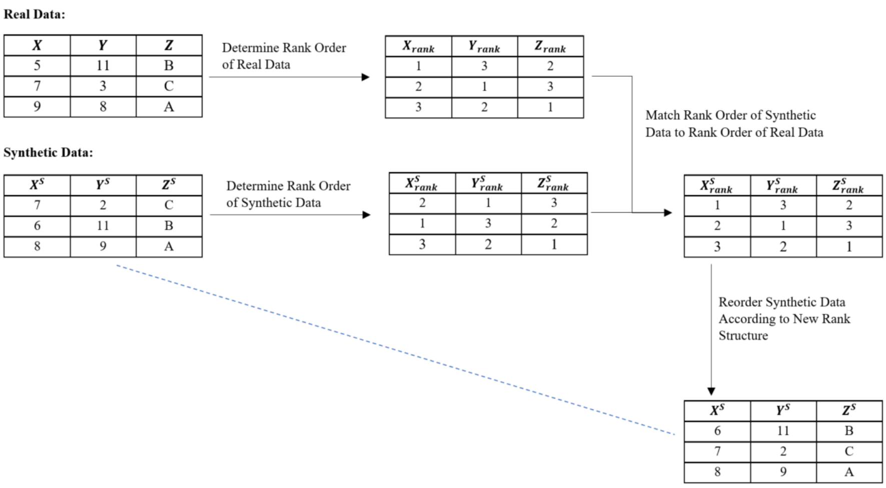


Figure 2: Example of Rank Matching Procedure used in `synthesizer`.

2.2.2 Expected Properties of the `sythesizer` Algorithm

Though relatively new, `synthesizer` implements an easy-to-understand method in a simple fashion. It is expected that the rank matching procedure in the algorithm will help to retain correlations between variables, and therefore the synthetic data will have similar overall distributional properties to the original data. Further, if there is no correlation in the original data, the expectation is that this procedure will not introduce it. Another expectation is that because of the rank-matching procedure, certain relationships in the data, for example, additive relationships such as $X + Y = Z$, will be replicated to some extent.

On the contrary, it is expected that the effect of rank-matching probably depends on the strength and form of the correlations and the number of variables involved. It is also still unclear how well linear and non-linear relations, as well as interactions, will be upheld by the method.

One of the aims of this thesis project is to test these expectations. This will give insight into the settings in which the method works well, as well as the situations in which it may yield unreliable results.

2.3 Benchmark Synthetic Data Generation (SDG) Method: Non-parametric `synthpop`

The `synthpop` package is a commonly used synthetic data generation (SDG) package and has outperformed its counterparts in literature [8]. Further, the package is intended for generating fully synthetic data. Thus, it will serve as a good baseline to compare `synthesizer` to.

When using the `synthpop` package, the user can decide between parametric and non-parametric methods. For the purpose of this thesis project, the non-parametric `synthpop` will be implemented, since it is the version of the package that performed best in the evaluation by Dankar et. al (2022) [8]. Additionally, since the `synthesizer` package also implements a non-parametric method, using the non-parametric `synthpop` package will allow for a more fair comparison.

The default non-parametric `synthpop` package uses classification and regression trees (CART) for data synthesis, implemented through the `rpart()` function from the `rpart` package [16]. The synthesis process begins by randomly sampling the first variable from the observed data, selecting a subset of the original values without considering relationships with other variables. Next, the second variable is generated using a CART model trained on the observed data, where the first variable serves as a predictor and the second variable as the target. The synthetic version of the first variable is then used to predict synthetic values for the second variable by sampling from the terminal nodes of the CART model. This method is repeated for the third variable, utilizing the first and second variables as predictors. The process continues iteratively, with each step incorporating additional predictors, until all variables are synthesized. For the final variable, all preceding variables are used as predictors. This sequential approach ensures that the relationships in the original data are largely maintained in the synthetic dataset. [17]

It is worth noting that the `synthpop` package has many options. It's main function comes with 22 settings, in addition to the input data and output size. For example, the visit sequence of the variables can be altered.

Chapter 3

Quality of Synthetic Data

For synthetic data to be an effective alternative to real data, it is important for synthetic data to be of good quality. The quality of synthetic data is multi-faceted and should take various aspects of the synthesized data into account. Specifically, the quality of synthetic data is not only determined by how similar the synthetic data is to the real data but also by how useful it is in the setting it was intended for, as well as by how well it holds up with regard to issues such as privacy and confidentiality.

There have been methods designed in literature to calculate a total quality score. These approaches focus on combining various individual metrics into a single comprehensive measure to evaluate synthetic data and compare synthetic different generation methods. For instance, Brenninkmeijer introduce the *Similarity Score (SS)*, while Chundawat et al. (2024) propose *TabSynDex* [18]. While these aggregated metrics provide a summary assessment of synthetic data quality, they sacrifice detailed insights into different facets of quality. Thus, most of the literature avoids using such metrics and continues to examine the quality of synthetic data through individual metrics that measure various aspects of quality, such as the distributional similarity between the original and synthetic data, the usefulness of the synthetic data in specific settings, and confidentiality. In this thesis project, individual measures will be prioritized to capture the distinct aspects of synthetic data quality comprehensively, with respect to RQ 1.

The literature emphasizes evaluating the usefulness of synthetic data when assessing synthetic data quality. Synthetic data is considered useful when inferences drawn from both original and synthetic datasets agree [19]. The usefulness of synthetic data can be evaluated in different ways. More specifically, the usefulness of synthetic data can be evaluated with respect to the distributional similarity between real and synthetic data [20] or with respect to a particular, narrowly-defined, purpose or application [21]. Some literature refers to both of these measures as utility measures. More specifically, these utility measures are referred to as general utility when looking at the distributional similarity, and as specific utility considering specific use cases.

On the contrary, some of the literature makes the distinction between fidelity and utility. This is because measuring the statistical similarity of synthetic and real data does not technically give insight into the 'usefulness' of the data, as 'usefulness' depends on the intended use-case of the synthetic data, as is described by the specific utility of the data. In this thesis report, the term fidelity [22] is used when describing and measuring the distributional similarity of datasets and specific utility when assessing the usefulness of the synthetic data in more specific settings.

It is also worth noting that data that is synthetic is not automatically always privacy-preserving, and may still face various disclosure risks. A disclosure refers to an instance wherein an individual or an enterprise is able to discern or learn something new about another individual or organization from released data [23]. A disclosure risk thus refers to the risk of identification or of uncovering private information and is therefore also a point of attention when assessing the quality of synthetic data.

Another concept related to the quality of synthetic data in the literature is information loss. Information loss examines the extent to which the synthesized data differs from the original data [24], such that there is little information loss if the structure of the two datasets is very similar. Literature oftentimes does not distinguish between information loss and fidelity (general utility). However, Taub et al. (2020) argue that the two concepts differ. This is because information losses in some variables may not necessarily impact the overall conclusions drawn from the data. Thus, the fidelity of the synthetic data may be preserved even when information is lost [25].

As the decisions people may want to make from the synthetic data are not predetermined, this thesis project will examine the 'usefulness' of synthetic data in terms that can be quantified, i.e. fidelity and specific utility. Further, due to time constraints, the privacy of the data synthesized will not be evaluated.

3.1 Fidelity

Although synthetic data is never intended to be an exact replica of the original data, it is important for it to exhibit distributional similarity to the original data [19]. A greater degree of distributional similarity between the original and synthetic data allows for more accurate statistical inferences to be drawn. The subsections below review some of the most commonly used metrics for assessing synthetic data fidelity and outline the metrics to be implemented in this thesis project.

3.1.1 Discriminant Based Metrics

Discriminant-based metrics, also called distinguishability metrics, assess the fidelity of synthetic data by measuring the extent to which it is distinguishable from the original data [8]. The main idea behind such metrics is thus to see how well some model can distinguish between real and synthetic data. Ideally, synthetic data should be similar to the real data, so the goal would be for a model to do a poor job of distinguishing between the two. Popular examples of such metrics in the literature include the propensity score measure [26], the propensity score mean-squared error (pMSE) ratio [27], and the *prediction MSE* [28].

Upon evaluating various distributional measures of general utility using both simulated and genuine data, Woo et. al (2009) found that propensity score methods are the most promising for estimating fidelity [26]. Since then, propensity score based measures, more specifically the pMSE have become famous for evaluating synthetic data in the literature, given its ease of use and advantages such as its ability to handle mixed-data types [19]. This thesis project will therefore use the *propensity score mean-squared error (pMSE)* to assess the distributional similarity between the original and the synthesized data.

The propensity-score algorithm starts by combining the original and synthetic data sets and assigning an indicator X to each record in the combined data, such that all rows of the synthesized data are assigned a value of 1 ($X = 1$) and all the rows of the original data are assigned a value of 0 ($X = 0$). The propensity score can then be thought of as the probability that a record in the combined data set is from the synthetic data, i.e. $P(X = 1)$. [26]

A binary classification model is then trained to predict the probability of each sample being synthetic or real [18]. Logistic regression or tree-based models are often the classification models of choice for calculating the *propensity MSE.* In this thesis project, a *Classification and Regression Tree (CART)* model is used to predict the probability of each sample being from the synthetic data.

CART is a widely used machine learning algorithm suitable for both classification and regression tasks. This algorithm belongs to the family of decision tree methods and works by recursively splitting the data into subsets based on the values of input features. The process results in a tree-like structure that can be utilized for making predictions [29].

The pMSE is then calculated, using the predictions from the CART model, as,

$$\text{pMSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{p}_i - 0.5)^2, \tag{3.1}$$

where $\hat{p}_i$ is the probability of each sample being either synthetic or real and n is the total number of samples after combining the synthetic and real data. The pMSE score ranges between 0 and 0.25, where lower scores indicate greater similarity between the real and synthetic data, and are thus preferred.

3.1.2 Exploratory Data Analysis Measures

Exploratory Data Analysis (EDA) refers to the process of performing initial inspections on data [30]. EDA is typically performed to discover patterns, identify anomalies, test hypotheses, and verify assumptions within the data. When assessing the fidelity of synthetic data, these aspects are often evaluated using graphical representations. Cluster analyses and pairwise correlation analyses can be performed by comparing the various original data plots to the synthetic data plots.

Woo et. al propose a cluster analysis measure to determine whether records in the original and synthetic data have similar values [26]. This is done using two plots of clusters (one for synthetic data and one for original data) and then checking if they look similar. Further, to evaluate how well feature interactions are preserved in the synthetic datasets, Zhao et. al (2021) compute the pair-wise correlation matrix for the columns within real and synthetic datasets individually [31]. The correlation matrices are then compared, and the differences between them should be small.

EDA will be conducted for the various original and synthesized datasets as part of the preliminary analyses in this thesis project. However, the results will not be included in this thesis project, as other fidelity metrics such as the pMSE provide a better numeric assessment of the fidelity.

3.1.3 Fidelity of Economic and Financial Data

Considering the nature of the economic and financial data in CBS' repository (zero-inflated with many missing values), and since the `synthesizer` package [7] was designed for synthesizing data of this nature, it is crucial to evaluate the method in this particular context. To evaluate the fidelity with respect to the presence of zero and missing values, the percentage of zero values and missing values in both the original and synthetic data should be compared. These percentages in the original and synthetic data should be close together. To get a numeric estimate of how close these percentages are for a dataset, the average mean squared error (MSE) is calculated. The average MSE quantifies the difference between the proportion of zeros or missing values in each variable of the original data and the corresponding proportion in the synthetic data. This metric will determine how effectively the syntheic data generation (SDG) method replicates the zero-inflation and missing values present in the original datasets.

The *Average Mean Squared Error (MSE)* between the zero or missing percentages of the real and synthetic datasets is calculated as,

$$\text{Average MSE} = \frac{1}{p}\sum_{j=1}^{p}\left(P_j^{\text{real}} - P_j^{\text{synth}}\right)^2 \tag{3.2}$$

where, p is the total number of variables in the dataset, P_j^{real} is the percentage for the j-th variable in the real dataset and P_j^{synth} is the percentage for the j-th variable in the synthetic dataset. A lower average MSE means the synthetic data more closely matches the real data in terms of the zero or missing percentages across variables. This suggests that the synthesis method effectively captures the zero-inflation or missing values in the original data. Therefore, lower average MSE values are preferred.

3.2 Specific Utility

It is important for synthetic data to not only share a similar underlying distribution with real data but also to serve its intended application effectively. Therefore, evaluating specific utility is crucial. Specific utility measures the 'usefulness' of synthetic data in relation to a particular, narrowly defined purpose or application [21].

Popular examples of specific utility measures include comparing summary statistics, such as means or variances, and calculating the standardized differences between these estimates in the original and synthetic datasets [32]. Another common approach involves comparing model coefficients derived from analyses, such as regression models performed on both original and synthetic datasets [19]. Confidence Interval Overlap (CIO), which measures the percentage overlap of confidence intervals for summary statistics or model coefficients, is also frequently employed to assess utility [19].

In this thesis, summary statistics are used solely for an initial assessment of the method and will not be included in the evaluation of the `synthesizer` method. Further, although the metrics mentioned above are effective for measuring specific utility, the focus of this thesis will be on evaluating the use of `synthesizer`'s synthetic data as training data for prediction purposes, as discussed in the next section. This decision is driven by time constraints.

3.2.1 Prediction Accuracy

Prediction accuracy is a popularly used specific utility metric [33], [34] and will also be used to measure specific utility in this thesis project. Dankar et. al (2022) [8] use the Train Synthetic Test Real approach (TSTR) [35] to compare the performance of models trained on synthetic and real data when tested on a real test set.

This approach begins with a real dataset, splitting it into a training dataset and a test dataset. A synthetic dataset is then generated using only the training data. Next, a model is trained twice, once using the synthetic data and once using the actual training data. Both models are evaluated on the test dataset, and their performance is compared. Comparing their performance gives a measure of how well the synthetic data retains utility for the specific modelling task. [36]

For this thesis project, either a logistic regression model or a multinomial logistic regression model will be trained using the above-described Train Synthetic Test Real (TSTR) approach, depending on the data type. Logistic regression is used to model the relationship between one or more independent variables and a binary outcome. It, therefore, estimates the probability of an event occurring based on a given data set of independent variables [37]. This type of statistical model is often used for classification and predictive analytics. Since the outcome is a probability, the predicted values range between 0 and 1, where values closer to 1 indicate a higher likelihood of the event happening and values closer to 0 indicate a lower likelihood. Multinomial logistic regression extends logistic regression to situations where the dependent variable has more than two categories.

Based on the logistic or multinomial logistic regression results, the F_1 score can be calculated to assess the quality of the synthetic data. F_1-scores provide a measure of predictive accuracy and will be implemented in this thesis project to evaluate the specific utility of the synthesized data. The F_1 score is a performance metric that balances the trade-off between precision and recall, providing a comprehensive measure of a model's effectiveness [38]. Precision refers to the accuracy of positive predictions, while recall quantifies the model's ability to identify all positive cases. For cases where there are only two classes, and logistic regression is implemented, the F_1 score is calculated as the harmonic mean of precision and recall, using the formula,

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall},} \tag{3.3}$$

where

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)},}$$

and

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN).}}$$

While, for the case where multinomial logistic regression is implemented, a one-vs-all classification approach is implemented. This means that each class is treated as a positive class, while all other classes are treated as negative [39]. An F_1 score is then calculated for each class using the above formula, and the scores for each class are then averaged into a single F_1 score.

The F_1 score ranges between 0 and 1, with higher scores indicating better model performance[38]. F_1 scores are particularly useful in assessing predictive accuracy when working with imbalanced data, as the F_1-score expresses the extent to which the model is correctly classifying the minority class [8]. When evaluating synthetic data, the goal is for the F_1 score of the synthetic data to be close to the F_1 score of the original data.

3.2.2 Domain Restrictions

One of the challenges of synthesizing economic or financial data is that such data often have logical and mathematical rules governing the allowable combinations of variable values. For example, a dataset may consist of three key variables, Cost (C), Profit (P) and Turnover (T), and the following relationship should hold for these variables: $C + P = T$.

Though the literature on synthesizing synthetic data in such a specific setting is limited, Taub et al. (2020) develop a *ratio of estimates (ROE)* for analyses involving totals or proportions [25]. The ROE is a simple metric that divides the smaller of the two estimates (i.e. the estimate from the original data or from the synthetic data) by the larger of the two estimates. For example, using the example from above, the metric may be calculated as $\frac{\text{Turnover(Real)}}{\text{Turnover(Synthetic)}}$.

Although this ratio effectively evaluates the accuracy of the final estimate, it does not account for whether the synthetic data preserves logical constraints, such as additive relationships. Building on the idea proposed by Taub et al. (2020), this thesis will adapt the metric to assess the individual components of the totals as ratios relative to the total

For example, using the example from above, $\frac{\text{Cost(Synthetic)}+\text{Profit(Synthetic)}}{\text{Turnover(Synthetic)}}$ will be compared to $\frac{\text{Cost(Real)}+\text{Profit(Real)}}{\text{Turnover(Real)}} = 1$. Ideally, these ratios in the synthetic data should equal 1, and thus values closer to 1 are preferred.

This analysis may serve as a possible alternative to an assessment of synthetic data by some sort of domain expert, as by checking such proportions, one would be mimicking an analysis by a domain expert to some extent.

Chapter 4

Methods

4.1 Experimental Design

The computational study is designed to assess the performance of the `synthesizer` package in a variety of settings, with respect to the metrics selected in chapter 3, in response to RQ 1. More specifically, the computational study will evaluate the performance of the `synthesizer` package against the framework of quality measures (RQ 2). Controlled experiments will, therefore, be conducted using both simulated data and real data. Three distinct data types will be considered: simulated multivariate normal data, simulated zero-inflated log-normal data and real survey data.

As outlined in RQ 3, to compare the `synthesizer` package with the established SDG package in R, the results from the experiments on all three data types will be compared to those obtained using the non-parametric `synthpop` package in R, enabling a direct comparison between the proposed method and an established approach. This comparison will provide a benchmark for evaluating the effectiveness of the proposed `synthesizer` method, highlighting any improvements or limitations in synthetic data quality relative to the widely used non-parametric `synthpop`.

The use of three data types ensures a thorough assessment of the method's performance in three different settings. Specifically, the utilization of simulated multivariate normal data facilitates testing the method in a perfect multivariate normal environment characterized by normally distributed data with no missing values or outliers. This creates an idealized baseline, enabling controlled testing of `synthesizer` under optimal circumstances. Employing simulated zero-inflated log-normal data that more closely resembles real data features, such as zero-inflation and a strong positive skew, then enables a performance assessment of the method in a more realistic setting. Nonetheless, this data is still free from any missing values, which often occur in real data, offering a realistic yet still simulated context. Finally, utilising real data enables a more precise evaluation of `synthesizer`'s performance in its intended application. This includes typical economic data that exhibit strong linear relationships, are zero-inflated, have numerous missing values and have mathematical restrictions (e.g. $X + Y = Z$). This combination of data types ensures a robust and comprehensive evaluation of the method, examining how the increasing complexity of data affects the quality of synthetic data generation.

4.2 Data Description: Simulated Data

To ensure a thorough assessment of `synthesizer`, the simulated data is set up to explore how dataset size and the distributional properties of data affect the quality of synthetic data generated using `synthesizer` and `synthpop`.

4.2.1 Simulation Conditions

Datasets with varying ratios of records to variables and varying distributional properties, such as different levels of correlation between variables and levels of separation between groups in the data, are simulated for the two distribution types, i.e., multivariate normal and zero-inflated log-normal. In each simulation, all simulated variables are numeric, with the exception of a single binary variable, which will also be simulated in each case. This binary variable will serve as the target variable for prediction.

Distribution Types

As mentioned in section 4.1, data will be simulated for two types of distributions, namely multivariate normal data and log-normal data. The log-normal data will be zero-inflated. The setup of the multivariate normal and the zero-inflated log-normal simulations are outlined in subsection 4.2.2 and subsection 4.2.3, respectively.

Separation Between Means

All datasets will have a total of $n = 1024$ records. To evaluate the performance of the `synthesizer` package on datasets of varying complexities, datasets will be simulated such that the first $\frac{N}{2} = 512$ records have a different mean than the second $\frac{N}{2} = 512$ records. The first $\frac{N}{2} = 512$ records are assigned a **True** binary label, and the second $\frac{N}{2} = 512$ records are assigned a **False** binary label. This True \ False label will be used for prediction. In each simulation, the difference between the means will be progressively reduced using various values, which will be detailed in the respective sections. By reducing the difference between the means, the level of overlap between the two halves of the dataset increases, increasing the complexity of the prediction task.

Correlation Between Variables

For each dataset half in the multivariate normal simulation, varying levels of correlation will be implemented. The setup will be such that the level correlation assigned will be the same between all variables. This will help explore how well the synthetic data generation (SDG) method can replicate data with increasingly complex relationships between variables in the datasets. The correlation levels will not be adjusted in the zero-inflated log-normal simulation, as the simulation for this distribution will be set up such that all variables are highly correlated.

Ratio of Records to Variables

To understand how changes in dataset scale and the ratio of records to variables affect the synthesis process, datasets with various ratios of records to variables will be simulated. Datasets with the following ratio of records to variables: 2:1, 4:1, 8:1, 16:1, 32:1, 64:1, 128:1, 256:1 and 512:1 will be simulated. Since $N = 1024$ is chosen for all datasets, datasets with $2, 4, 8, 16, 32, 64, 128, 256$ and 512 variables will be simulated to create the aforementioned ratios for each combination of parameters. This is done because, as the number of variables rises, the potential multivariate effects increase combinatorially.

4.2.2 Multivariate Normal Data

Datasets with numeric variables following a multivariate normal distribution are simulated, as the consistent and predictable properties of this distribution provide an ideal testing environment and a controlled benchmark for evaluating the `synthesizer` algorithm.

To achieve the aforementioned separation between the means, two data frames will first be generated separately for each simulation. The first data frame will have $\frac{N}{2} = 512$ records and a positive mean of μ. The second data frame will have $\frac{N}{2} = 512$ records and a negative mean of $-\mu$. All variables in each data frame are normally distributed, with a mean μ or $-\mu$ and a variance of $\sigma^2 = 1$. To decrease the level of separation between the means, three μ values are considered, $\mu = \{2.5, 1.25, 0.625\}$. The variance will be set to $\sigma^2 = 1$ for all simulations. The correlation between all variables in each data frame will be set to r.

Figure 3 illustrates the variance-covariance matrix for a dataset half with p variables, where the variances are represented on the diagonal, and the correlation r is the same between all variables. Note, since $\sigma = 1$ for all variables, correlation = covariance for all datasets.

$$
\Sigma = \begin{array}{c|ccccc}
 & x_1 & x_2 & x_3 & \cdots & x_p \\
\hline
x_1 & 1 & r & r & \cdots & r \\
x_2 & r & 1 & r & \cdots & r \\
x_3 & r & r & 1 & \cdots & r \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
x_p & r & r & r & \cdots & 1
\end{array}
$$

Figure 3: Covariance matrix structure for $x_1, x_2, \ldots, x_p$, where the diagonal contains variances which were set to 1, and all off-diagonal elements represent the correlation r between variables.

The correlation will be the same between all variables of a dataset, i.e. all variables in a single dataset will be equally correlated. However, to examine how different strengths of correlation between variables affect synthetic data quality, datasets will be simulated for different r values.

Datasets with correlation levels of $r = \{0, 0.25, 0.5, 0.75\}$ will be simulated, ranging from no correlation to strong correlation.

The two data frames are then merged into a single dataset with $N = 1024$ records. All the datasets will also have a True/False column. The first $\frac{N}{2} = 512$ records with μ are assigned a True label, while the second 512 records with $-\mu$ are assigned a False label. This True/False column serves as the target variable for prediction in this setting. All other variables in the dataset are used as predictors to model and predict this binary outcome.

A visualization for datasets with 2 variables and $r = 0$ is shown for the three different μ values in Figure 4.

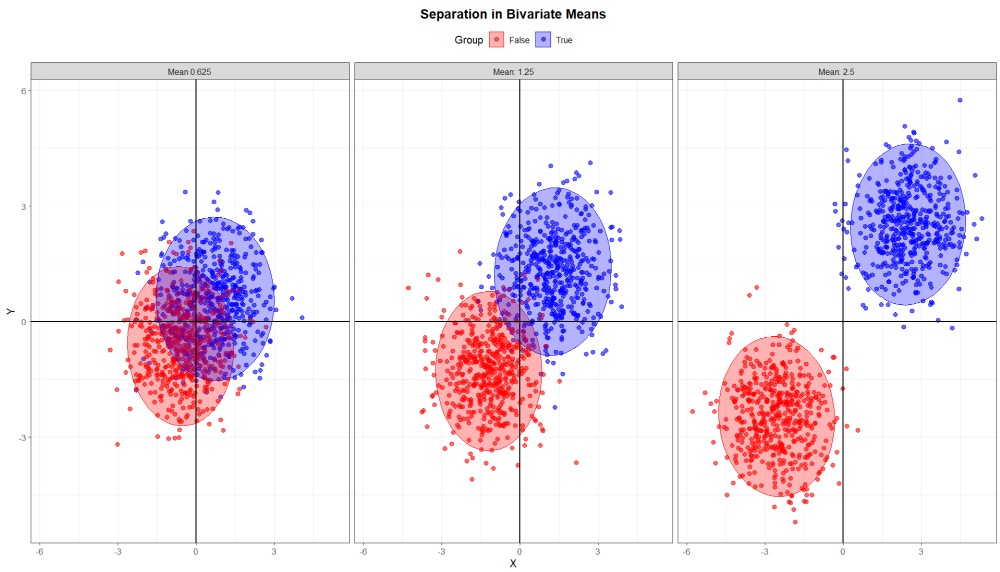


Figure 4: Plot showing an example of the separation between groups for a generated dataset with 2 variables with $r = 0$. The blue points are from the first half of the dataset with mean μ and a 'True' label. The red points are from the second half of the dataset with mean $-\mu$ and a 'False' label. Notice how for as μ gets smaller, the groups overlap more. This overlap is exacerbated when the variables are correlated.

Multivariate normal data will be generated for all ratios of records to variables (9 conditions), correlations (4 conditions) and means (3 conditions). Therefore, $9 \times 4 \times 3 = 108$ multivariate normal datasets will be simulated. Table 1 shows an example of one of the possible dataset simulations for the case of two variables.

	$X1$	$X2$	**True/False**
1	0.826	0.971	True
$\vdots$	$\vdots$	$\vdots$	$\vdots$
512	0.343	0.047	True
513	-1.866	-2.495	False
$\vdots$	$\vdots$	$\vdots$	$\vdots$
1024	-2.383	-2.998	False

Table 1: Example of a generated dataset for the multivariate normal simulation with 2 variables $X1$, $X2$ i.e. a 512:2 ratio of records to variables. The first 512 records have a mean of 1.25, while the next 512 have a mean of -1.25. The correlation for this dataset was $r = 0.25$ True/False column designed for prediction is also included.

4.2.3 Zero-Inflated Log-Normal Distribution

Zero-inflated log-normal data is generated to assess the `synthesizer` algorithm in a more realistic scenario than the multivariate normal distribution. This type of data better represents the distribution frequently observed in economic variables, which are often zero-inflated, right-skewed, and strictly positive.

The setup of the zero-inflated log-normally simulated data resembles that of the multivariate normal data by merging two data frames, which are simulated with different means, while the gap between these means is systematically narrowed.

The data is simulated such that for each half of the data set, the values of each variable are drawn from a log-normal distribution with a mean of μ_1 assigned to the first 512 records and a mean of μ_2 assigned to the second 512 records. For both dataset halves, the variance is set to 1, $\sigma^2 = 1$. Datasets will be created for values of $\mu_2 = \{6, 3.5, 2.25\}$ while μ_1 will remain set to 1 for all simulations. This will bring the means of the two groups closer to one another, as in the previous simulation.

To introduce correlation between the variables, each subsequent variable in a dataset half is generated by adding a small amount of random noise to the values of the previous variable. This noise is drawn from a standard normal distribution with a mean of zero, $\mu = 0$ and a standard deviation $\sigma = 1$, allowing for a controlled degree of variability and inter-variable dependence.

The two data frames are then merged into a single dataset with $N = 1024$ records. As in the multivariate normal simulation, all the datasets will also have a True/False column. The first $\frac{N}{2} = 512$ records with $\mu_1 = 1$ are assigned a True label, while the second 512 records with μ_2 are assigned a False label. To introduce zero inflation, 10% of the numeric values in the data are randomly assigned a value of zero. This allows for the creation of synthetic datasets that mimic real-world data characteristics, namely strong correlations between variables and zero inflation.

Table 2 gives an example of such a possible dataset with 4 variables with $\mu_1 = 1$ and $\mu_2 = 6$.

Datasets with this setup will be generated for all ratios of records to variables (9 different parameters) and μ_2 values (3 parameters). Therefore, $9 \times 3 = 27$ zero-inflated log-normal datasets will be simulated.

	X1	**X2**	**X3**	**X4**	**True/False**
1	0.85	1.30	0.94	1.05	True
2	0.75	1.22	1.02	1.15	True
⋮	⋮	⋮	⋮	⋮	⋮
512	0.90	1.45	1.10	0.95	True
513	6.00	6.10	6.20	6.05	False
⋮	⋮	⋮	⋮	⋮	⋮
1023	6.05	6.00	6.10	6.00	False
1024	6.25	6.20	6.10	6.15	False

Table 2: Example of a synthetic dataset with 4 variables and 1024 records. The first half (records 1-512) have a mean of $\mu_1 = 1$, and the second half (records 513-1024) have a mean of $\mu_2 = 6$.

4.3 Data Description: Real Data

To test the `synthesizer` package on real economic data, this thesis project utilizes real data from the 2007 Structural Business Survey (SBS). The SBS is an annual survey conducted by Statistics Netherlands (CBS) to assess the structure, conduct, and performance of economic activities in the Netherlands. It includes data collected from all non-financial business sectors, excluding agriculture and personal services, with variables collected at the economic sector level [6]. Thus, as part of the SBS there are many datasets, each of which corresponds to a different sector.

The raw datasets include varying numbers of records and variables, where financial variables are reported in thousands of euros [6]. Not all variables in the datasets provide unique or relevant information. For example, several administrative variables, such as the ID variable, are not informative for predictive purposes. These administrative variables are excluded from the datasets to refine the data for analysis. After excluding these variables, most variables in the data exhibit varying degrees of missingness. The initial data analysis also revealed that many variables are subject to several linear restrictions and inequalities, and most are heavily right-skewed and strictly positive.

Further, each dataset contains a categorical variable `GK` between 0-9 that categorises businesses according to size. This variable is used for prediction in the real data.

Ten datasets are selected for analysis to evaluate datasets of different sizes. Table 3 shows the number of records, variables, and ratio of records to variables for each selected dataset.

Data Set	**Records**	**Total Variables**	**Records:Variables**
1	4836	94	51.4:1
2	7250	61	118.9:1
3	2938	112	26.2:1
4	2665	82	32.5:1
5	2473	77	32.1:1
6	1135	93	12.2:1
7	1263	83	15.2:1
8	1201	72	16.7:1
9	161	90	1.8:1
10	276	62	4.5:1

Table 3: Properties of the selected real datasets, namely the number of records, number of variables and the ratio of records to variables (rounded to 1 decimal) for each dataset.

4.4 Synthetic Data Quality

To evaluate the performance of the `synthesizer` package, the quality of the synthetic data it generates needs to be assessed. This is done through various metrics. As detailed in chapter 3, the selected evaluation metrics will be implemented to assess the quality of the synthetic datasets. Not all metrics will be applied to every dataset; rather, they will be implemented based on the characteristics of the datasets under consideration, namely the simulated multivariate normal, simulated zero-inflated log-normal, and real data. Each dataset type has unique attributes that dictate the applicability of specific metrics. Table 4 gives an overview of the selected metrics, their formulas, and the datasets to which they will be applied.

Metric	Formula	Multivariate Normal	Zero-Inflated Lognormal	Real Data
Propensity Mean Squared Error (pMSE)	$\frac{1}{N}\sum_i (\hat{p}_i - 0.5)^2$	✓	✓	✓
Percentage of Zero Values	$\frac{\text{Count of zeroes}}{\text{Total records}} \times 100$ $\text{Average MSE} = \frac{1}{p}\sum_{j=1}^{p}\left(P_j^{\text{real}} - P_j^{\text{synth}}\right)^2$		✓	✓
Percentage of Missing Values	$\frac{\text{Count of missing values}}{\text{Total records}} \times 100$ $\text{Average MSE} = \frac{1}{p}\sum_{j=1}^{p}\left(P_j^{\text{real}} - P_j^{\text{synth}}\right)^2$			✓
F_1 **Score**	$2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$	✓	✓	✓
Ratios	$\text{RATIO}_{\text{not on payroll}} = \frac{\text{temporary workers + other temporary staff + other persons}}{\text{total employed persons not on payroll}}$ $\text{RATIO}_{\text{total}} = \frac{\text{total employed persons not on payroll + total employed persons on payroll}}{\text{total employed persons}}$			✓

Table 4: Overview of quality assessment metrics, their formulas, and their applicability to multivariate normal, zero-inflated lognormal, and real data.

4.4.1 Implementing the Quality Metrics

The selected metrics are implemented following the formulas and procedures outlined in chapter 3. However, the implementation details for the specific utility F_1 scores and ratio metrics, particularly as applied to the real data, will be provided in the subsequent sections.

F_1 Score

In all simulations and for the real data, a train-test split of 70:30 will be used. This means that 30% of the real data will be reserved for testing the models trained on the simulated and real data.

The real data does not have a True/False column, and the `GK` is used for prediction in the real data. The `GK` variable is used to classify businesses based on their size, and it is a categorical variable with values 0-9. Therefore, multinomial logistic regression is implemented as opposed to logistic regression. Prior to implementing the multinomial logistic regression, all variables where more than 80% of records are missing will be excluded, as they are deemed useless in training models.

Further, the trained models are unable to be tested on the real data due to the presence of missing values. The missing values in the real test data will thus imputed. Imputation will be done using k-nearest neighbours, with $k = 5$. This means that each missing value will be assigned a value equal to the average of the five observations closest to it.

Ratios

Two main variables are selected from the real data to implement the ratio metric designed in chapter 3. These variables are *total employed persons not on payroll* and *total employed persons*, where *total employed persons not on payroll* is a subtotal of *total employed persons*. More specifically,

total employed persons not on payroll = temporary workers + other temporary staff + other persons

and

total employed persons = total employed persons not on payroll + total employed persons on payroll,

where total employed persons on payroll = persons on payroll + persons loaned out.

The ratios for the two variables will thus be calculated as,

- $\text{RATIO}_{\text{not on payroll}} = \frac{\text{temporary workers + other temporary staff + other persons}}{\text{total employed persons not on payroll}}$ and
- $\text{RATIO}_{\text{total}} = \frac{\text{total employed persons not on payroll + total employed persons on payroll}}{\text{total employed persons}}$

These variables are selected for the ratios, as they appear in all 10 real datasets, to allow a consistent comparison of the ratios across the different datasets. Note, records in the real data for which this relationship is violated (where the ratio does not equal 1 in the real data) will be removed prior to synthesis.

4.5 Synthetic Data Generation

The synthetic data generation method is easily implemented in R using the `synthesize` function from the `synthesizer` [7] package, which implements Algorithm 1. The R package will be utilized in its default configuration. To implement the non-parametric `synthpop` in R, the `syn` function from the `synthpop` library is used, with `method = "cart"`, specifying the implementation of the CART approach to synthesizing data.

Since `synthesizer` implements a random sampling procedure as outlined in Algorithm 1, the quality of the data synthesized may vary for different iterations and may be much better for one iteration, than for another. Therefore to reduce the impact of this randomness, quality metrics will be calculated for more than one synthesized dataset and averaged for a more stable estimate of the quality. To determine the appropriate number of synthetic datasets to be generated from a single original dataset, an iterative experiment was implemented on one of the real datasets. In each iteration, an additional synthesized dataset was added to the existing set, and the average propensity mean squared error (pMSE) across all synthesized datasets was recalculated. This updated pMSE was compared to the previous iteration's pMSE. The process was repeated for 50 synthesized datasets. In an attempt to reduce the randomness as well as balance computational efficiency a threshold of 0.01 was selected. As shown in Figure 5, 14 datasets were synthesized before the change in pMSE between successive iterations remained below the threshold of 0.01. As an extra precaution, it was decided that for each data set, all metrics will be averaged over 15 synthesized data sets.

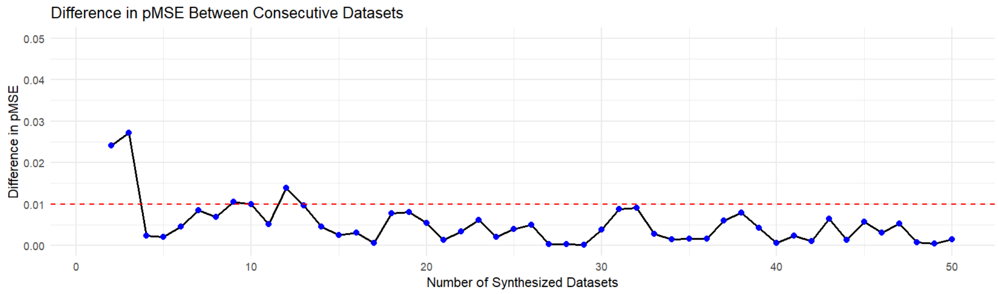


Figure 5: Plot showing the difference in average pMSE between consecutive synthesized datasets, synthesized using the `synthesizer` package. Used to identify when the pMSE difference becomes small enough to indicate convergence in the synthesis process.

Chapter 5

Results

This chapter presents the results of the computational study, organized by data type: simulated multivariate normal data, simulated zero-inflated log-normal data, and real datasets. The analysis focuses on evaluating the performance of the proposed method, `synthesizer` against the benchmark method `synthpop` with reference to the behaviour of synthesized datasets under varying conditions using the selected quality measures. These results provide insights into the strengths, limitations, and patterns that influence the effectiveness of the methods across different scenarios. Note, all metrics in the results section are averaged over 15 synthetic datasets.

5.1 Simulated Multivariate Normal Data

To evaluate the quality of the data synthesized from the simulated multivariate normal datasets, the pMSE and F_1 scores are compared across all simulated multivariate normal datasets.

Propensity Mean Squared Error (pMSE)

The propensity mean squared error (pMSE) evaluates a model's ability to distinguish between real and synthetic data, where lower pMSE values indicate greater similarity between the real and synthesized data. pMSE values are bounded between 0 and 0.25 and values closer to zero are preferred.

Figure 6 illustrates the pMSE (averaged over 15 synthetic datasets) for each combination of simulation conditions. The x-axis denotes the number of variables in the simulated data, where a higher number of variables corresponds to a lower ratio of records to variables (since there are 1024 records in all datasets), and the y-axis represents the average pMSE. The rows correspond to the three different μ values, indicating the mean of the normal distribution and, hence, the level of overlap between the two groups in the data. Recall that each dataset consists of two halves with means μ and $-\mu$. The columns distinguish between the synthetic data generation (SDG) method implemented (`synthesizer` and `synthpop`). The different shades of the lines in the plots reflect the different levels of correlation between the variables in the simulated dataset.

For synthesizer, pMSE values increase as the mean values rise. This indicates greater differences between the real and synthesized data when there is more separation between the groups in the data, i.e. there is less overlap between the means. The observation is thus that synthesizer performs worse when there is greater separation between the means of the dataset halves. This issue is exacerbated in the absence of correlation, particularly when there are more than 128 variables (when the ratio of records to variables is less than 8:1). This is shown by the increased pMSE for the lightest line in the 1st row of plots for the synthesizer method. A potential explanation for this is that when the data exhibits complete separation, the inverse eCDF is not able to capture this separation, as values between the two groups may still be sampled. This could be further examined by calculating the pMSE after separately synthesizing the two halves of the data and then combining them. Another possibility is that when the data has two separate groups and there is no correlation between the variables, rank matching becomes ineffective as there is no relationship between the variables. The rank-matching might therefore serve no real purpose in the absence of relationships between variables.

Contrarily, synthpop exhibits lower pMSE values when the means are less overlapped (for greater μ values). This suggests that synthpop outperforms synthesizer in conditions where the two groups in the simulated data have greater separation. This could be because this simulation design of separate groups in the data benefits a CART model. When a dataset has two distinct dataset halves, a CART model can divide the synthesis into two independent problems during the initial split. Subsequently, the CART model then more effectively replicates the two halves in the real data.

With the exception of the scenario discussed where $\mu = 2.5$ and the number of variables exceeds 128, the pMSE for synthesizer generally rises with higher correlation levels, showing that performance decreases when the variables in the data are more correlated. Whereas for synthpop, the opposite trend emerges, as pMSE tends to decrease with increases in correlation values, meaning that performance increases as correlations increase. This pattern might be explained by synthpop's synthesis process, which iteratively adds predictors to synthesize variables. This sequential approach helps maintain relationships present in the original data within the synthetic dataset, especially in the presence of highly correlated variables. Whereas for synthesizer, while the rank matching does preserve the structure of the data, the method is not as optimized for correlations of greater than 0.75 as synthpop, and may be more suited to more moderate correlations.

For both SDG methods, a greater number of variables is consistent with increased pMSE and, therefore, worse performance. A greater number of variables means that there is a lower ratio of records to variables in the data since the number of records is held constant for all simulations. Thus, this outcome is not surprising, as any synthetic data method requires enough examples to understand the relationships among variables to generate realistic synthetic data.

F_1 Score

In this thesis project, the F_1 score is employed as a measure of specific utility to assess the performance of synthesized data as training data for predictive purposes, compared to the original data. The F_1 score ranges between 0 and 1, with higher values indicating better predictive accuracy. The primary focus in the context of synthetic data is, however, on determining whether the F_1 scores obtained using synthesized data are comparable to those using the actual data. If the F_1 scores of the synthetic data are closely aligned with those of the simulated data, it suggests that the synthetic data closely preserves the essential structure and relationships of the simulated data, making it a reliable substitute for model training. Therefore, the goal is for the F_1 score of the synthesized data to be close to the F_1 score of the simulated data.

Figure 7 compares the F_1 scores (averaged over 15 synthetic datasets for each method) across three different training sets: the simulated data, data synthesized using `synthesizer`, and data synthesized using `synthpop`. All three models were tested on a subset of the original data, following the Train-Synthetic-Test-Real (TSTR) approach outlined in chapter 3. In the plot, the x-axis represents the number of variables and the y-axis shows the F_1 score. The rows correspond to the different mean (μ) values of the normal distributions, and different shades indicate varying levels of correlation.

Generally, as expected, the results indicate that as the means in the simulated multivariate normal (MVN) datasets become less separated (i.e. smaller μ values) and correlations increase, the F_1 scores tend to decrease, reflecting the growing difficulty in distinguishing between classes. However, due to the simplicity of the simulated MVN data, the scores remain relatively high across all iterations. When the μ values are further apart, the classification problem becomes trivial, with minimal overlap between classes. Consequently, the F_1 scores are higher, often reaching perfect values due to the complete separation in the data.

When examining the F_1 scores across the methods, for the different simulation conditions, `synthpop` more closely tracks the overall patterns exhibited by the F_1 scores in the simulated data compared to `synthesizer`. It also often happens that data synthesized using `synthesizer` produces higher F_1 scores than the simulated data. This may be because models trained on simulated data are prone to overfitting, whereas the synthetic data generated by `synthesizer` introduces more variations that encourage the model to generalize better to unseen data. Whereas, `synthpop`'s ability to more closely replicate the F_1 scores of the real data could be attributed to `synthpop`'s ability to generate synthetic data that more often closely mimics the original simualted data based on the lower pMSE scores just seen in the previous subsection. Therefore, it appears that there is a relationship between pMSE values and differences in F_1 scores, as lower pMSE values seem to correspond to smaller differences in pMSE values between the simulated and synthesized data. This idea is further reinforced when looking at the specific simulation conditions. When $\mu = 2.5$, the F_1 scores for the `synthesizer` method are on the upper boundary (close to or equal to 1) and sometimes higher than the simulated data values, while the `synthpop` method depicts a more similar pattern to the simulated data. In the previous subsection, when the means were more separate, the pMSE scores for `synthesizer` were higher (indicating worse performance) than for `synthpop`.

This trend persists in the correlations; as the correlations increase, the divergence between the F_1 scores of the simulated data and the `synthesizer` method also increases. In contrast, this divergence primarily occurs with `synthpop` when the correlation is zero. This further illustrates a relationship between the pMSE scores of the methods and their closeness to the F_1 scores, as in the previous subsection `synthesizer` had higher pMSE values for greater correlations, while the opposite was true for `synthpop` which had lower pMSE values for greater values or correlation.

This also explains why when the number of variables is 128 or more, there seems to be a more pronounced divergence between the F_1 scores of the methods and the simulated data. More specifically, both methods experienced a decline in performance (higher pMSE values) for a greater number of variables (smaller ratios of records to variables).

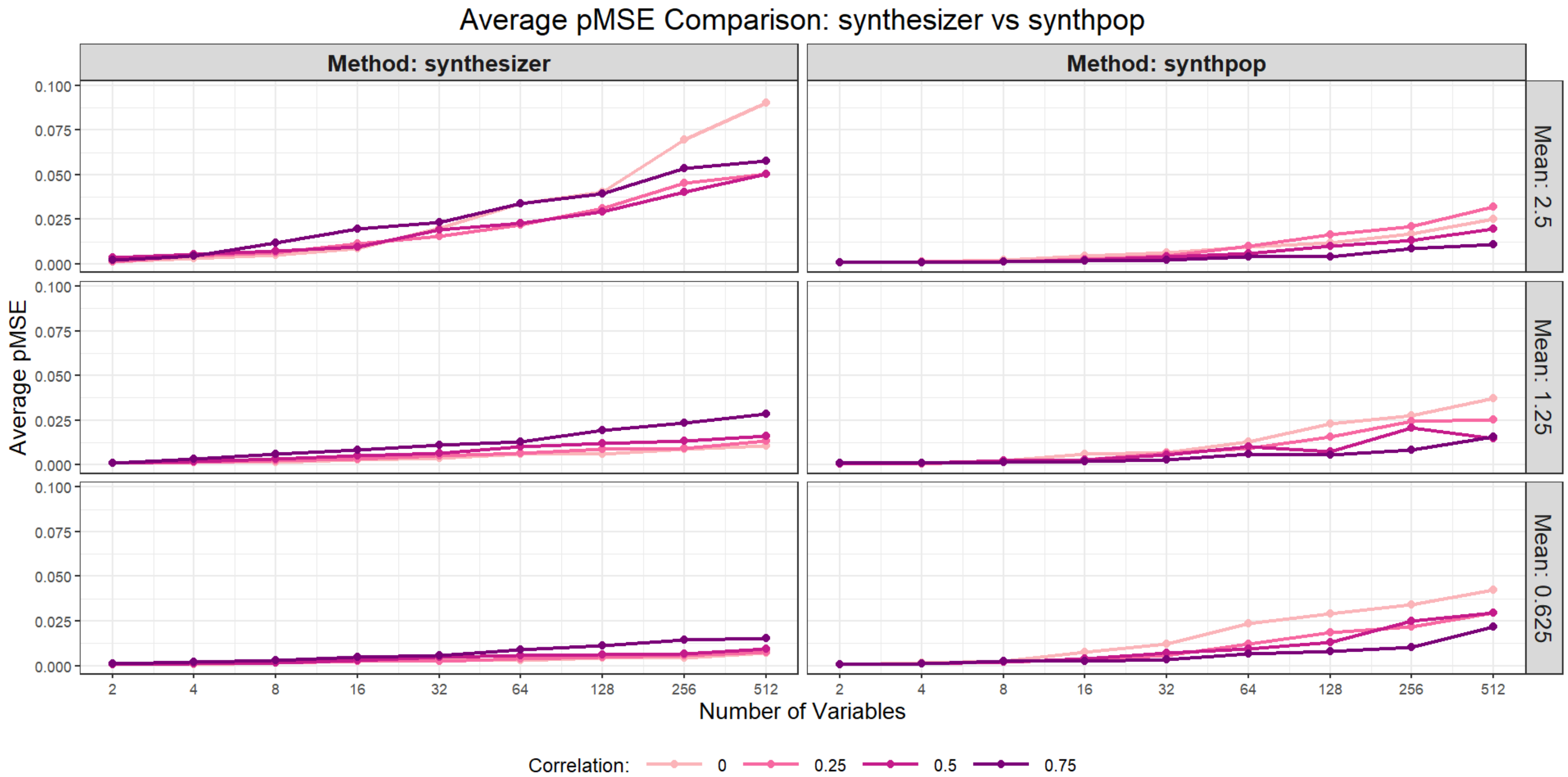


Figure 6: Propensity mean squared error (pMSE) results (averaged over 15 synthetic datasets) for the various multivariate normal simulation conditions (means, correlations and ratios of records to variables) are provided for the `synthesizer` and `synthpop` synthetic data generation (SDG) methods. pMSE values are bounded between 0 and 0.25. Smaller pMSE scores indicate that the synthetic data is more similar to the real data, and are therefore preferred.

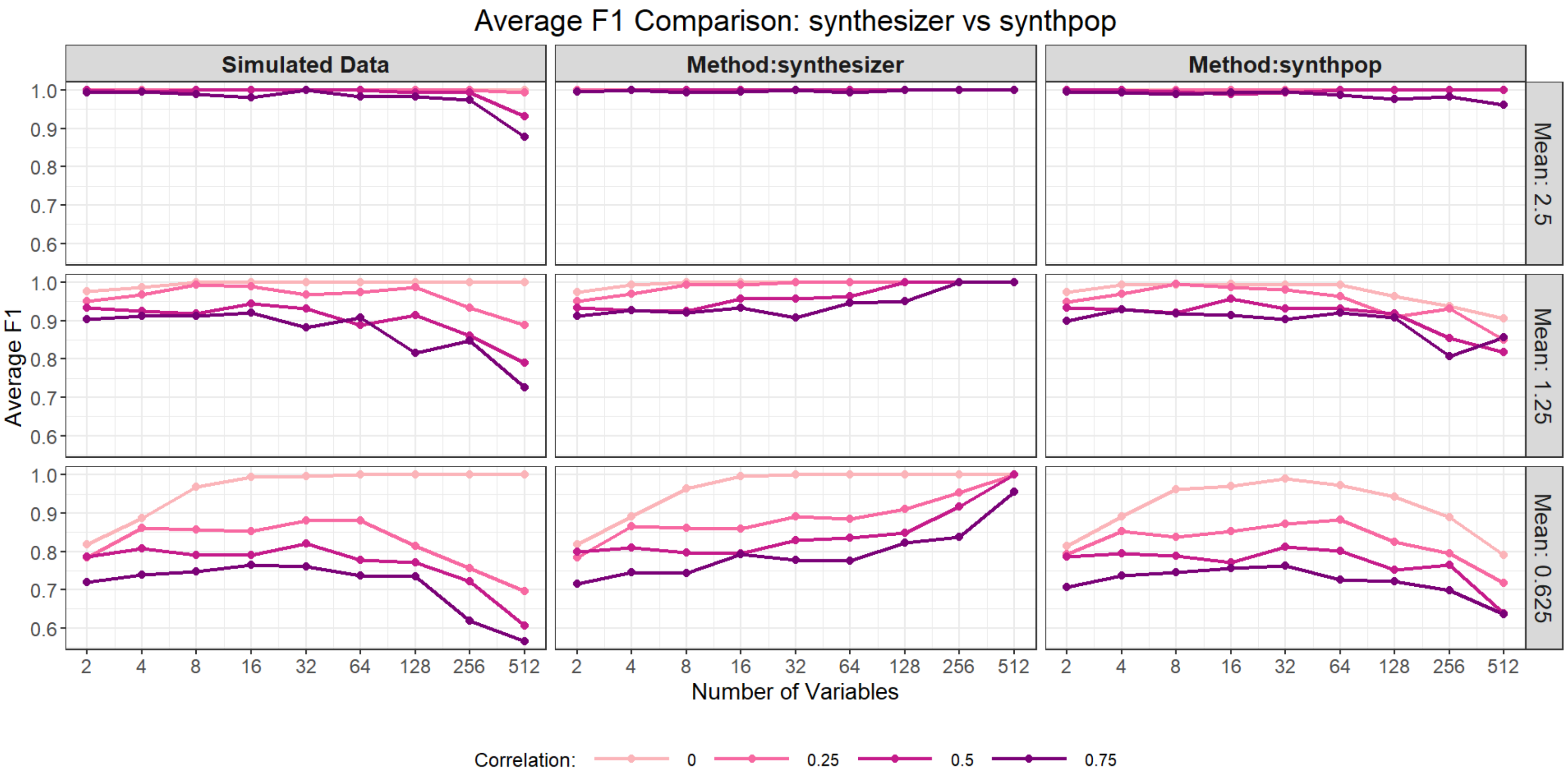


Figure 7: F_1 results for the various multivariate normal simulation conditions (means, correlations and number of variables) are provided for the `synthesizer` and `synthpop` synthetic data generation (SDG) methods, as well as the original simulated data. F_1 scores are averaged over 15 synthetic data sets for both methods. F_1 values are bounded between 0 and 1, and higher F_1 scores indicate better predictive performance. When evaluating the synthetic data generation (SDG methods), F_1 scores should be compared to those of the real data. Scores closer to the real data values indicate that the synthetic data more closely mimics the real data.

5.2 Simulated Zero-Inflated Lognormal Data

To assess the quality of the synthesized data in a context that more closely mimics economic data i.e. the simulated zero-inflated log-normal datasets, the pMSE, F_1 scores, and percentage of zero values are compared across all simulated zero-inflated log-normal datasets.

Propensity Mean Squared Error (pMSE)

The pMSE values were calculated for all synthesized zero-inflated lognormal conditions. and are presented in Figure 8.

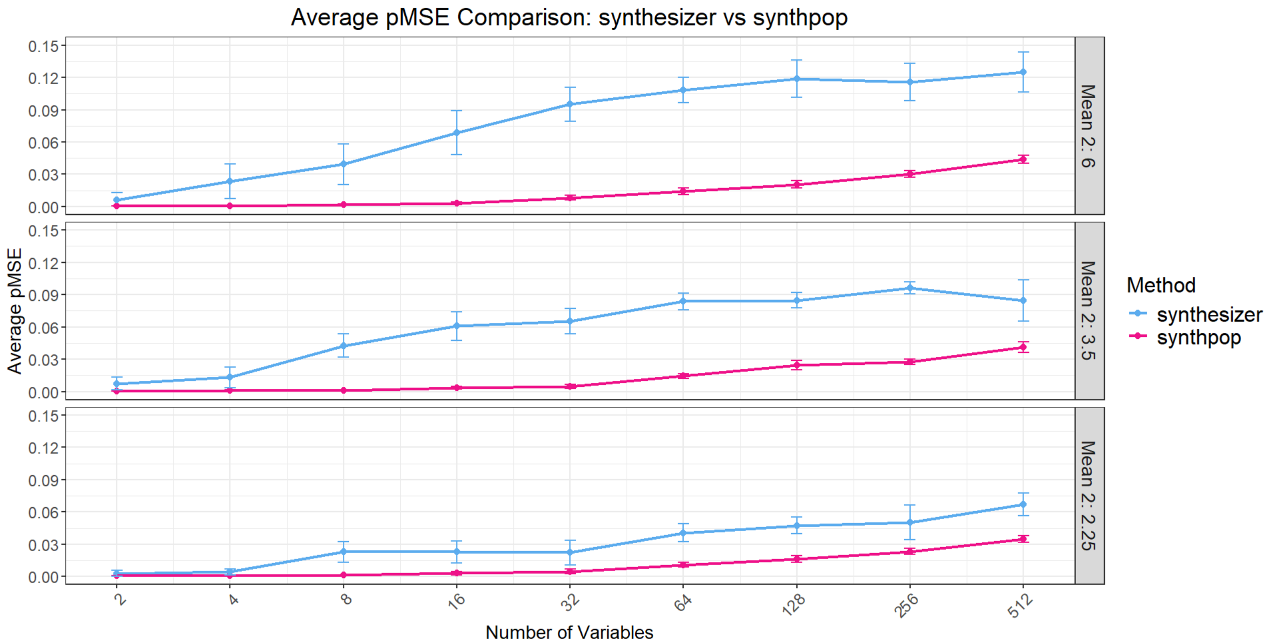


Figure 8: Propensity mean squared error (averaged over 15 synthetic datasets) results for all zero-inflated log-normal simulation conditions (means and ratios of records to variables) presented for `synthesizer` and `synthpop` synthetic data generation (SDG) methods. pMSE values are bounded between 0 and 0.25, and smaller pMSE scores are preferred since they indicate that the synthetic data is more similar to the simulated data.

The x-axis displays the number of variables (where more variables represent a smaller ratio of records to variables) and the y-axis displays the average pMSE (averaged over 15 synthetic datasets). Lower pMSE values indicate higher-quality synthetic data. Recall that this simulation also had two groups within each simulated dataset, $\mu_1 = 1$ and μ_2, and that no correlation conditions were included in this simulation, but that variables were highly correlated. The rows represent the three different μ_2 values, with larger values indicating greater separation between the means. The different colours in the plot represent the two synthetic data generation (SDG) methods, `synthesizer` and `synthpop`.

For this data type, `synthpop` has lower pMSE values compared to `synthesizer` across all simulation conditions. Recall that lower pMSE values mean better performance, therefore showing that `synthpop` consistently outperforms `synthesizer` in this simulation. This is likely due to the high correlation between variables, highlighting `synthpop`'s superior ability to capture strong linear dependencies and effectively handle heavily correlated variables, due to its iterative setup. As mentioned in the previous section, while the rank-matching implemented in `synthesizer` is able to preserve relationships between variables, it is not as optimised for highly correlated variables as `synthpop`. Another point worth mentioning is that the way in which the data is simulated in this setting may favour the `synthpop` algorithm, in that the data is simulated such that each variable is a combination of the previous variable plus some noise. This may benefit `synthpop`, as it learns from one variable in the data to predict the next, then uses those two variables to predict the next variable, etc., and therefore may more accurately replicate the setup of the data.

As in the multivariate normal data simulation, `synthesizer`'s performance declines (as seen by higher pMSE values) as the means become further separated, underscoring its difficulty in handling complete separation within the data. This further reinforces that `synthesizer` struggles in situations where the groups in the data are completely separate. As mentioned earlier, this may be because by sampling from the inverse eCDF it is likely that values between the two groups may be sampled in the `synthesizer` algorithm, as opposed to `synthpop`, which implements a CART model that can easily identify these groups at the initial split.

Additionally, a noticeable increase in pMSE is also observed again for both methods as the number of variables rises (and the ratio of records to variables declines), reinforcing the idea that both methods see a decline in the quality of synthetic data when there are fewer records per variable in the data. Having fewer records for each variable negatively impacts the ability of the methods to learn patterns in the data and relationships between variables, therefore negatively impacting the quality of the synthesized data.

F_1 Score

Using the same setup as in the previous multivariate normal simulations, the F_1 scores are calculated. Recall that when the F_1 scores of the synthesized data are closer to the F_1 scores of the original simulated data, it means that the synthetic data more closely replicates the simulated data. The calculated F_1 scores (averaged over 15 synthetic datasets) are presented in Figure 9. The number of variables are shown on the x-axis (with more variables representing smaller ratios of records to variables), and the F_1 scores are shown on the y-axis. The different μ_2 values are represented in the rows, and different colours are used to distinguish between the two methods and the simulated data.

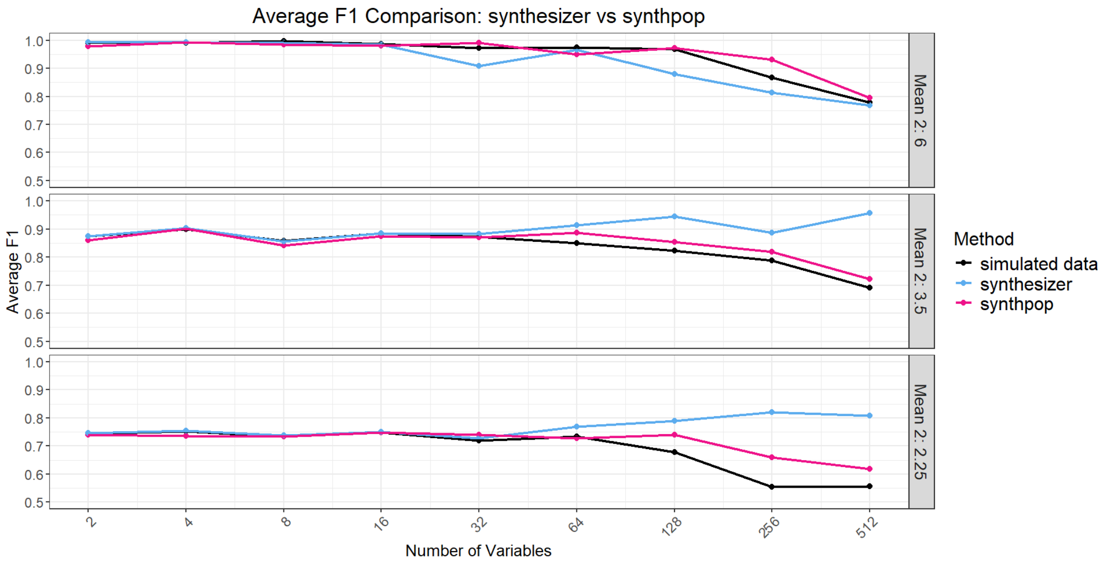


Figure 9: F_1 score results (average over 15 synthetic datasets for both methods) for all zero-inflated log-normal simulation conditions (means and number of variables) presented for `synthesizer` and `synthpop` synthetic data generation (SDG) methods. F_1 values are bounded between 0 and 1, and higher F_1 scores indicate better predictive performance. When evaluating the SDG methods, F_1 scores should be compared to those of the original data. Scores closer to the simulated data values indicate that the synthetic data more closely mimics the original simulated data.

When $\mu_2 = 2.25$ and $\mu_2 = 3.5$ and the data has more than 32 variables, both the synthetic data generated using `synthesizer` and `synthpop` yield a higher F_1 score than the real data. However, the F_1 scores of `synthpop` are closer to the real data F_1 scores than those of `synthesizer`. The higher F_1 scores seen by `synthesizer` likely correspond to the pMSE values in the previous subsection. It appears that when pMSE values are slightly higher, and the synthetic data isn't as close to the real data, it serves as a better training data set, hence `synthesizer`s increased F_1 score. However, this is only true to a certain extent until the opposite effect occurs. This is seen when $\mu_2 = 6$, as `synthesizer`'s F_1 score drops below the others here. This is likely because `synthesizer` experienced a deline in performance (higher pMSE values) here, as `synthesizer` struggles when there is greater separation between the halves of the data (as seen in previous subsections). Whereas `synthpop`'s F_1 scores follow the simulated F_1 scores more closely in this setting, up until there are 128 variables or more, because as mentioned before, the CART model can better replicate this separation between the halves of the data.

Also, both methods follow the F_1 scores of the simulated data very closely, up until there are 16 variables for $\mu_2 = 6$, and up until 32 variables for the other mu_2 values. Since a greater number of variables corresponds to a smaller ratio of records to variables in this setting, this reiterates the fact that both methods see a decline in performance when there are not sufficient records in relation to the number of variables in the data.

Percentage of Zero Values

To assess the extent to which `synthesizer` can replicate a common attribute in economic and financial datasets, namely zero inflation, the percentage of zero values for all variables in both the real and synthesized datasets was calculated. The goal is for the percentage of zero values for a variable in the synthesized dataset to closely match the percentage of zero values in the real data.

Figure 10 presents the results for a simulated dataset with 512 variables and $\mu_2 = 3.5$. Results are similar for other datasets with this number of variables. Plots for all other datasets in this simulation are available in Appendix A, section A.1. The percentage of zero values in the simulated dataset are represented on the x-axis, and the percentage of zero values in the synthetic data on the y-axis. Each dot represents a variable in the datasets, and the figure distinguishes between `synthesizer` and `synthpop` using different colours. The dashed line represents the line where $y = x$, i.e. the line where the zero percentage in the synthetic data equals the zero percentage in the simulated data. Points closer to this line are better, as it means the values in the synthesized and real data are closer to each other.

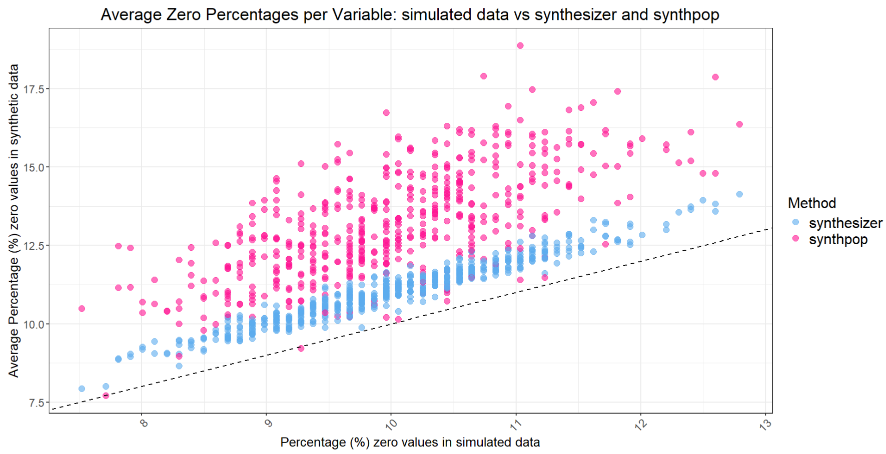


Figure 10: Zero percentage results for zero-inflated log-normal simulation setting where $n = 512$ and $\mu_2 = 3.5$. Each point in the plot represents a variable in the data. Points closer to the dashed line indicate that the percentage of zero values in the synthesized data is closer to the percentage of zero values in the simulated data and are thus are preferred.

For both methods, the dots form a linear pattern above the dashes lined, indicating that the methods are overestimating the percentage of zero values. The reason for this overestimation of zero percentage by both methods is not completely clear, but it could be due to the way in which data was simulated, where zero percentages were introduced to 10% of the data.

Although the zero percentages generated by `synthesizer` do not exhibit a perfect linear relationship with the zero percentages in the simulated data and tend to be slightly higher than the dashed line, the points for `synthesizer` method are closer to the dashed line than the points of the `synthpop` method. Therefore, `synthesizer` provides a more accurate estimation of the zero percentages in the data. This is attributed to `synthesizer's` sampling approach, in which zeroes are sampled in proportion to their occurrence for each variable, whereas `synthpop` takes more of an iterative approach to identify patterns in the data, which might not replicate the zero percentages as accurately.

To get a better estimate of how closely the percentage of zero values for each method is to the percentage of zero values in the simulated datasets, the average mean squared error was calculated between the percentages in the simulated data and the percentages in the synthesized data for each dataset. These MSE results are presented in Figure 11.

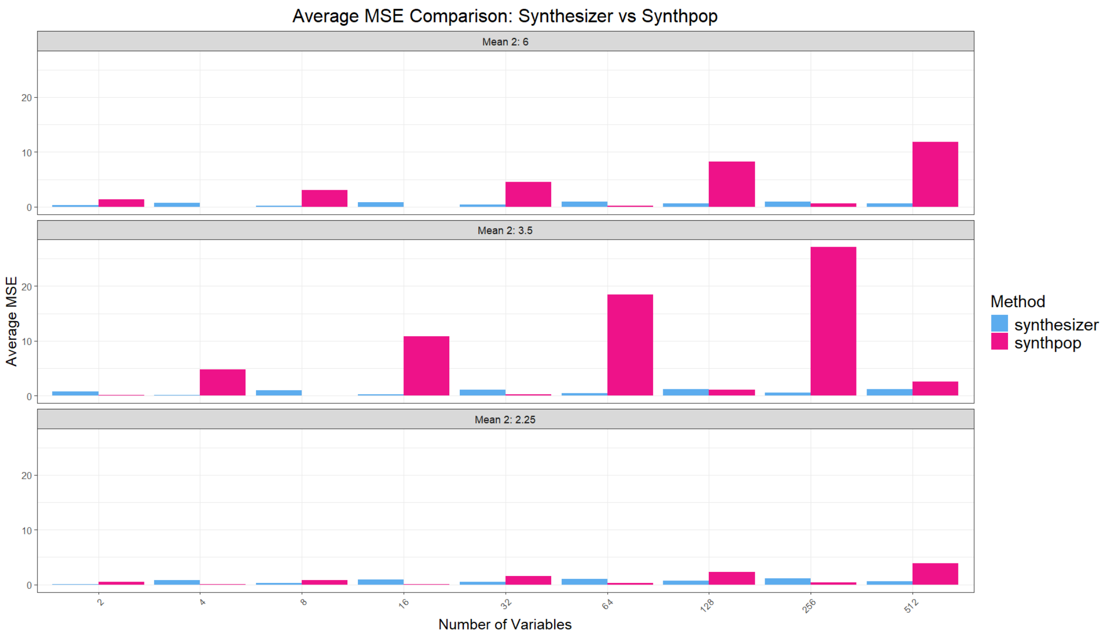


Figure 11: MSE of zero percentage results for zero-inflated log-normal simulation setting for all simulation conditions (μ_2 values and ratios of records to variables). Smaller MSE values indicate that the percentage of zero values in the synthesized data is closer to the percentage of zero values in the simualted data and are thus preferred.

The plot displays the number of variables on the x-axis, and the average MSE of the zero percentages (averaged over 15 datasets) on the y-axis. The rows distinguish between the different μ_2 values. The two methods, namely `synthesizer` and `synthpop` are represented in different colours. Recall that lower MSE values indicate that the average difference between the zero percentage in the simulated data and the synthesized data is smaller and are therefore preferred.

For most cases, `synthesizer` has a lower MSE than `synthpop`. This is likely because `synthesizer`'s method means that zero values get sampled in a way that is proportional to their occurrence in the actual data, as opposed to `synthpop`, which has to assign zero values based on patterns learned from the data. Therefore, it makes sense that `synthesizer` outperforms `synthpop` with respect to this metric. However, there are some instances when `synthpop` outperforms `synthesizer` (lower MSE), for example when there are 256 variables and $\mu_2 = 6$ and $\mu_2 = 2.25$. The reason for this is unclear, and there seems to be no explicit pattern in the instances in which `synthpop` performs better. When `synthpop` does outperform `synthesizer`, the differences appear small, therefore it may just be that sometimes the patterns learned by `synthpop` provide a slightly more accurate estimate of the zero percentages than explicitly sampling these zero values in proportion to their occurrence. It could also be that there are too few iterations of the data being synthesized and that `synthpop` is outperforming `synthesizer` by chance in these settings. This remains unclear.

5.3 Real Data

The quality of data synthesized from the real data is thoroughly evaluated using five key metrics: pMSE, F_1 scores, percentage of zero values, percentage of missing values, and two ratios ($\text{RATIO}_{\text{not on payroll}}$ and $\text{RATIO}_{\text{total}}$) which check whether the synthesized datasets preserve the additive relationships in the real data.

Propensity Mean Squared Error (pMSE)

To assess the distributional similarity between the synthesized and real data, the pMSE is evaluated. Figure 12 presents these results. The real dataset numbers are shown on the x-axis, and the average pMSE (averaged over 15 synthetic datasets) is shown on the y-axis. Note that datasets with smaller indices tend to have larger ratios of records to variables (see Table 3). Each method is represented by a different colour. The mean pMSE values are represented by dots, with error bars representing the 95% confidence intervals (CIs).

The results show no clear pattern, with `synthpop` outperforming `synthesizer` in some scenarios and vice versa. For datasets 2 and 4, the pMSE is lower for `synthpop`, indicating more distributional overlap with the original datasets compared to the `synthesizer`. This is likely due to the increased linear relationships in larger economic datasets (those with a greater number of records to variables), which `synthpop` has shown to be really good at replicating in comparison to `synthesizer`, as emphasized in the previous simulations. It is therefore likely that `synthpop` is able to better utilize the larger dataset sizes to learn patterns more effectively compared to `synthesizer`.

However, for the smaller datasets (6, 7, 8, 9 and 10), the error bars overlap a lot, and the performance of the metrics align more closely. Additionally, for smaller datasets 8, 9 and 10, the confidence intervals are wider for both methods, indicating greater uncertainty in the results. This is possibly because the patterns in these datasets are less pronounced, preventing `synthpop` from learning them as clearly or effectively, and therefore it does not perform that much better than `synthesizer`.

For some datasets, e.g. datasets 2 and 5, `synthesizer` had a lower pMSE than `synthpop`. However, there remains no clear indication of when which method does better, and it is unclear why `syntheiszer` outperformed `synthpop` in these datasets.

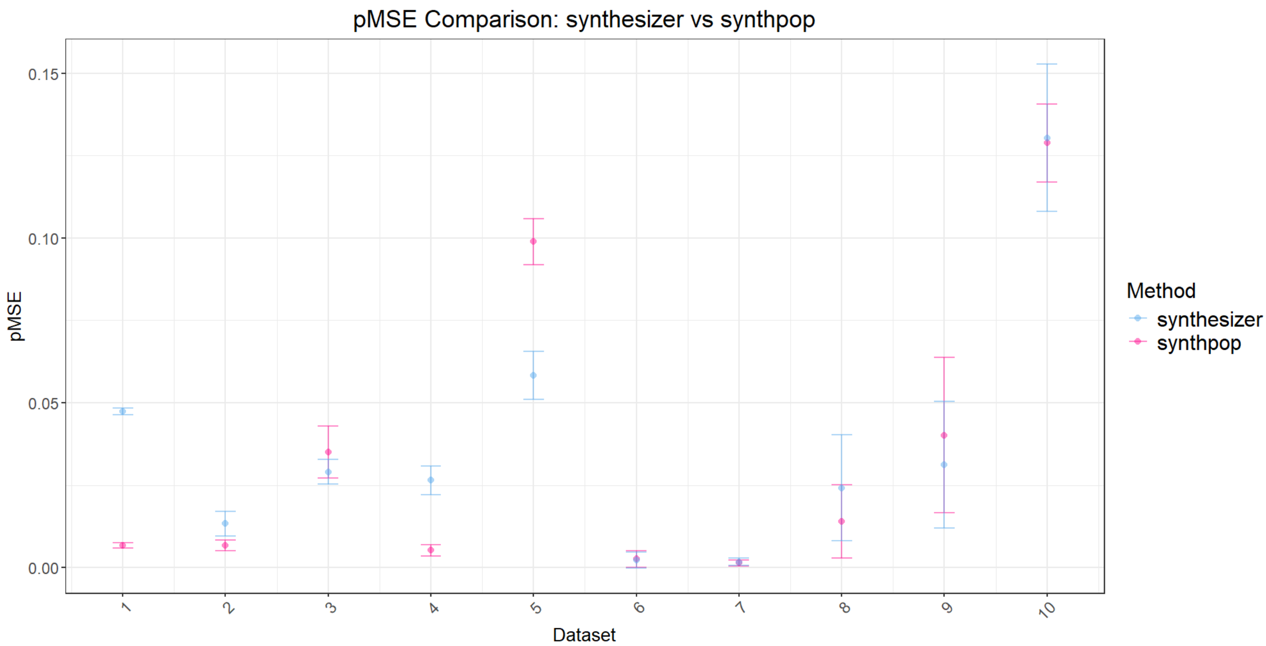


Figure 12: Propensity mean squared error results (averaged over 15 synthetic datasets) for the real data setting are provided for the `synthesizer` and `synthpop` synthetic data generation (SDG) methods. pMSE values are bounded between 0 and 0.25. Smaller pMSE scores indicate that the synthetic data is more similar to the real data, and are therefore preferred. Also note: datasets with smaller indices tend to have bigger ratios of records to variables

F_1 Score

The F_1 analysis of the real data compares the performance of synthesized datasets as training data for prediction purposes to the real data.

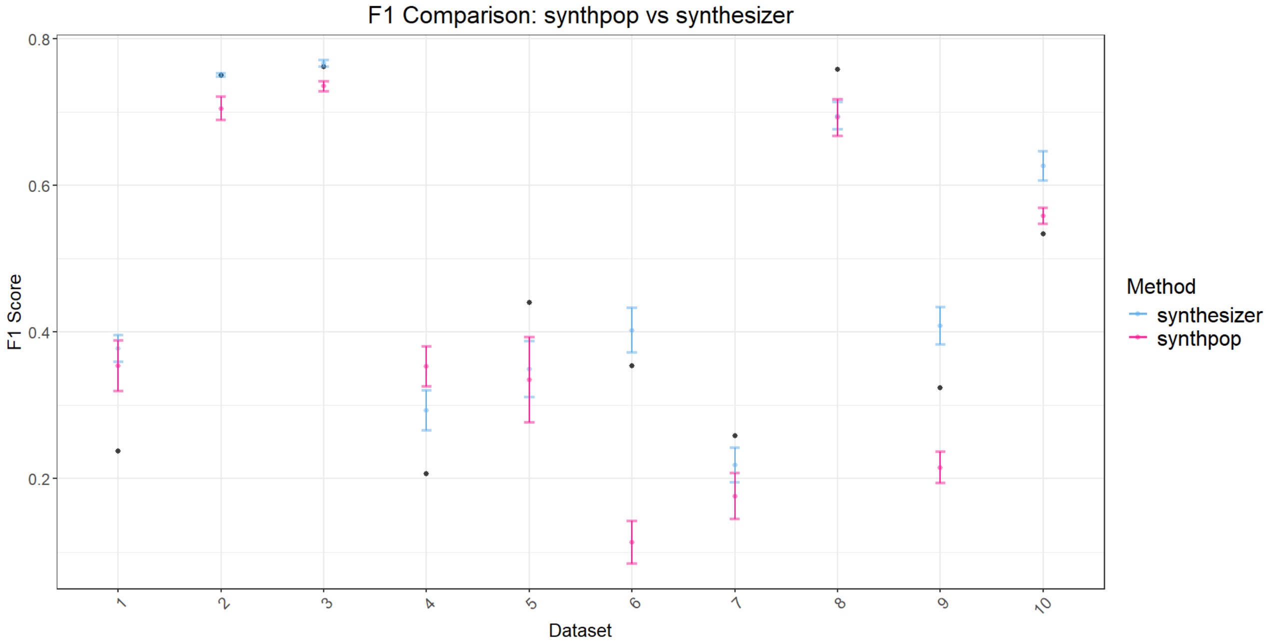


Figure 13: F_1 score results for 10 real economic datasets presented for `synthesizer` and `synthpop` synthetic data generation (SDG) methods. F_1 scores for original datasets are shown as black dots. F_1 values are bounded between 0 and 1, and higher F_1 scores indicate better predictive performance. When evaluating the SDG methods, F_1 scores should be compared to those of the real data, scores closer to the real data values indicate that the synthetic data more closely mimics the real data.

The results of this analysis are presented in Figure 13, where the dataset indices are shown on the x-axis and the F_1 score on the y-axis. Again recall that datasets with smaller indices tend to have larger ratios of records to variables. Each method is represented by different colours. F_1 scores for the real data are represented by the black dots. The average F_1 scores for both methods are represented by dots, with error bars representing the 95% confidence intervals (CIs).

In this analysis, `synthesizer` performs better than `synthpop` for most datasets (2-8) with F_1 scores closer to the real data F_1 scores. A possible explanation for this is again that by mimicking the data slightly less accurately `synthesizer` introduces variations that allow the model to generalize better, whereas the real data might lead to overfitting in some cases. `synthpop`, on the other hand, may not perform as well because its synthetic data generation method more closely mimics the real data, potentially making it less effective in training models that generalize well.

Percentage Zero Values

Comparing the percentage of zero values in real and synthesized data helps assess how well the synthesized data replicates the percentage of zero values in real data. In this analysis, the percentage of zero values in the real datasets is compared with the corresponding percentages in the synthesized data.

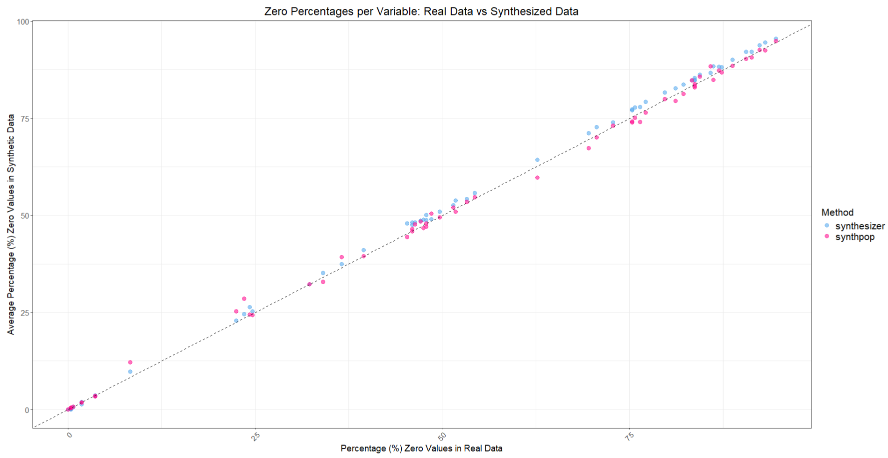


Figure 14: Zero Percentages in Real and Synthesized Datasets using both `synthesizer` and `synthpop` for real dataset 10. Each dot represents a variable in the data. Ideally, a perfect linear relationship between percentages in the real and synthesized data should be observed, such that points fall on the grey dashed line.

The results for dataset 10 are shown in Figure 14. The percentage of zero values in the real data is represented on the x-axis and the percentage in the synthesized data on the y-axis. A linear relationship should be observed, indicating that the zero values in the synthesized datasets closely match those in the real data. Ideally, these points should be exactly on the $x = y$ line.

The results show that both methods perform well and exhibit a linear relationship with the true values. The remaining plots, which can be found in the Appendix A, section A.2, tell a similar story, further supporting the overall trend observed in these analyses.

Similar to the previous section, to better estimate how closely the percentage of zero values for each method aligns with the percentage of zero values in the simulated datasets, the average mean squared error (MSE) is computed between the percentages in the real data and those in the synthesized data for each dataset. These MSE results are presented in Figure 15.

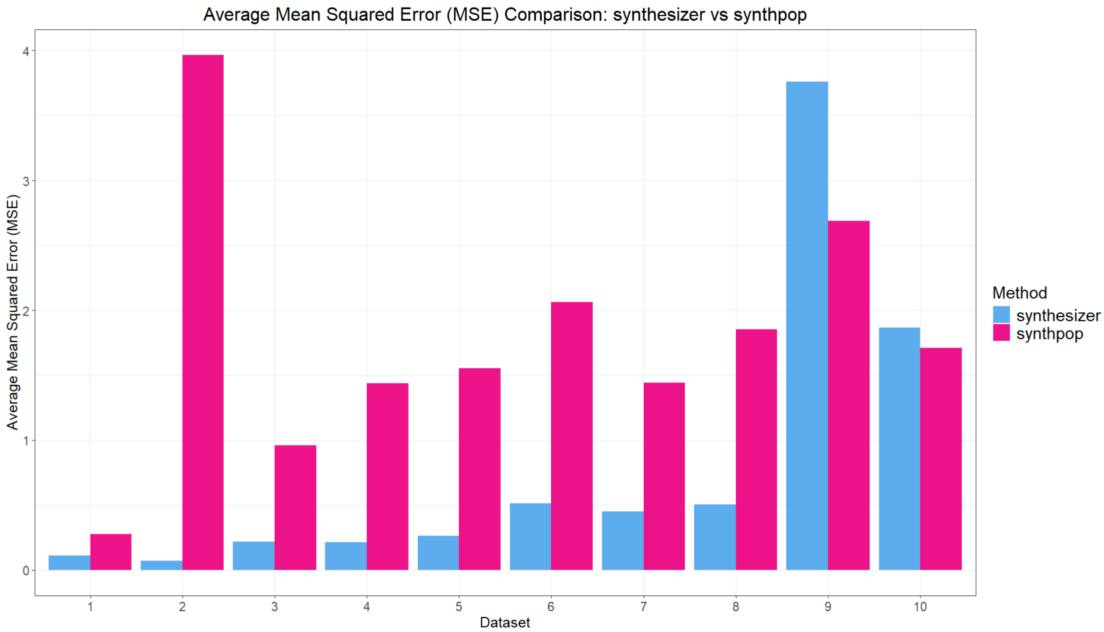


Figure 15: MSE of zero percentage results for real datasets. Smaller MSE values indicate that the percentage of zero values in the synthesized data is closer to the percentage of zero values in the real data and are thus preferred. Note: datasets with smaller indices tend to have a larger ratio of records to variables.

The results show that `synthesizer` has a lower MSE for 8 out of the 10 datasets, meaning it more closely replicates the percentage of zero values in the real data for 8 out of 10 of the datasets, with the exception of the last two small datasets. As mentioned in the previous section, `synthesizer` does well at replicating zero values in the data, as it samples zero values in proportion to their occurrence, rather than having to assign zero values based on data patterns learned.

Also as mentioned in the previous section, it is again not completely clear why `synthpop` is outperforming `synthesizer` in some settings, as in the last two datasets here. However, since in this instance, `synthpop` is outperforming `synthesizer` for the two smallest datasets, a further evaluation could be done to see if the occurence of zero values between variables are highly correlated for these smaller datasets, as if this is the case, it would make sense that `synthesizer` does a good job at replicating zero values; as it's iterative procedure is able to replicate relationships for highly correlated variables well.

It is worth noting that the 95% confidence intervals (CIs) for the percentage of zero values were also examined. However, further analysis revealed that although the point estimates across the synthetic datasets are generally close, the convergence of the distributions of zero percentages does not occur across the 15 synthesized datasets. This indicates that more datasets would need to be synthesized for the full range of zero percentage distributions to converge and for the CIs to be reliable. The same holds true for the confidence intervals of the percentage of missing values.

Percentage Missing Values

The percentage of missing values in datasets is an important attribute for assessing the quality of synthesized economic data. This metric evaluates how effectively the synthesized data mirrors the distribution of missing values in real data.

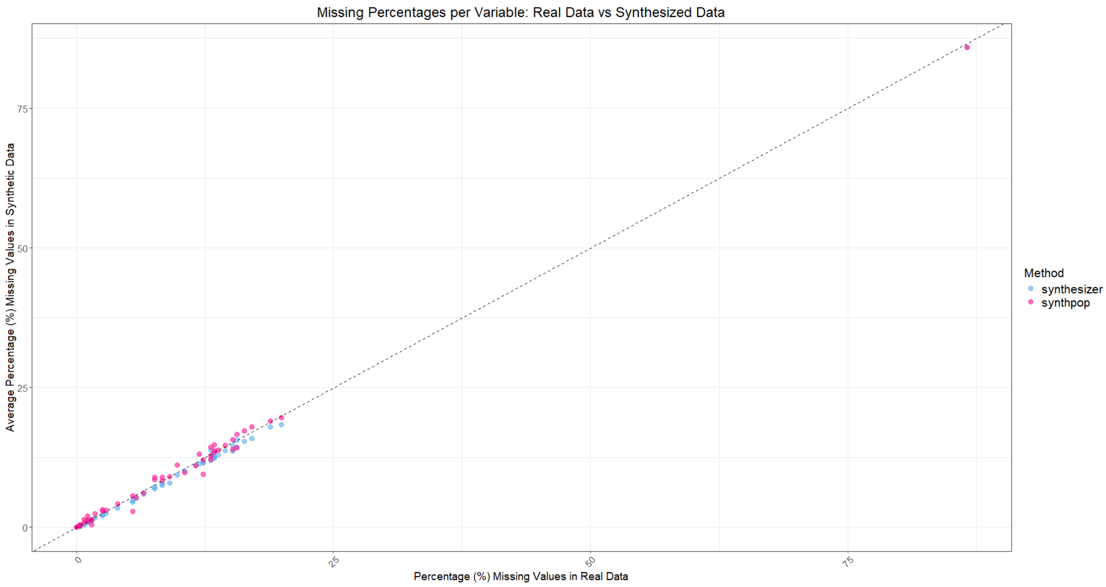


Figure 16: Missing Percentages in Real and Synthesized Datasets using both `synthesizer` and `synthpop` for real dataset 10. Each dot represents a variable in the data. Ideally, a perfect linear relationship between percentages in the real and synthesized data should be observed, such that points fall on the grey dashed line.

These results for real dataset 10 are displayed in Figure 16 with the percentage of missing values in the real data plotted on the x-axis and the percentage in the synthesized data on the y-axis. Ideally, the points should lie along a diagonal line, indicating a perfect linear relationship between the real and synthesized data. The two methods are represented in different colours.

The results show that `synthesizer` demonstrates a slightly more linear and precise relationship compared to `synthpop`, though both methods perform reasonably well. This difference is likely due to `synthesizer` incorporating missing values in a more direct manner by sampling them in proportion to their occurence in the real data, while `synthpop` attempts to infer patterns, which may not always replicate the exact distribution of missing values. The remaining plots, provided in Appendix B, reveal similar trends, reinforcing these observations across other datasets.

As done for the percentage of zero values, to better assess how closely the percentage of missing values for each method aligns with the real datasets, the average mean squared error was calculated between the missing value percentages in the real and synthesized data for each dataset. These MSE results are presented in Figure 17.

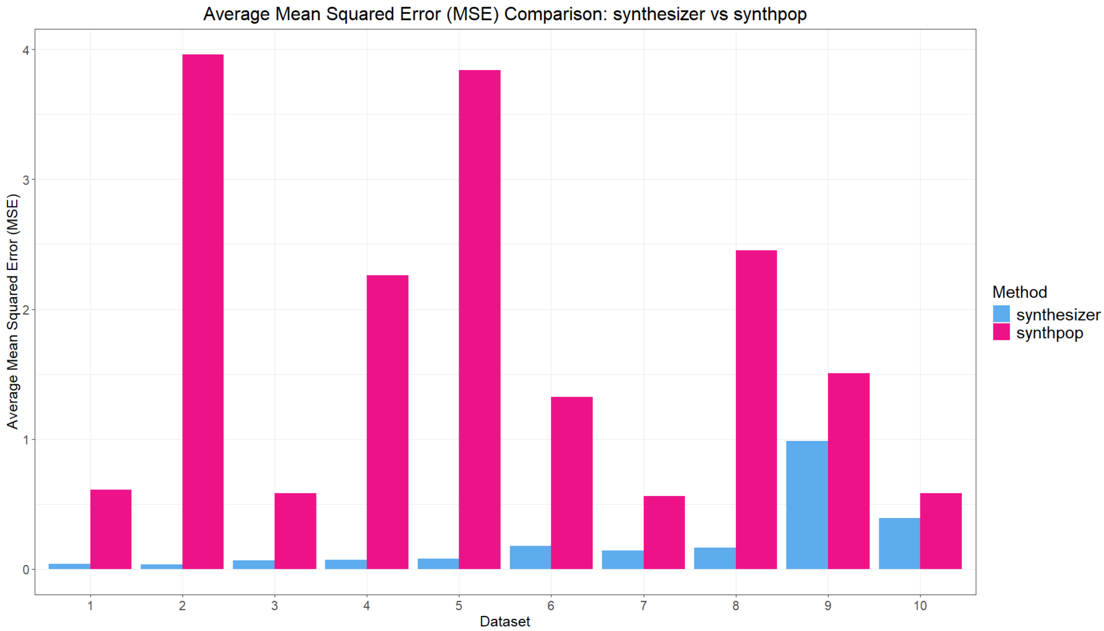


Figure 17: MSE of missing percentage results for real datasets. Smaller MSE values indicate that the percentage of missing values in the synthesized data is closer to the percentage of missing values in the real data and are thus preferred. Note: datasets with smaller indices tend to have a larger ratio of records to variables.

In this setting, `synthesizer` has lower (sometimes even much lower) MSE values for all datasets compared to `synthpop`. This is because `synthsizer` deals with missing values in a similar fashion to zero values, in that they are sampled in proportion to their occurrence in the real data. It appears that `synthpop` is struggling to replicate missing values. It may be that the occurrence of missing values is less or not correlated between variables, meaning that they would be occurring somewhat "randomly" and that might explain why `synthpop` is struggling to learn the patterns in which they occur.

5.3.1 Ratios

To evaluate whether additive relationships are preserved in the synthesized data, the ratios of the components of a total to the actual total are analyzed. Specifically, this assessment focuses on the total number of employed persons not on the payroll $\text{RATIO}_{\text{not on payroll}}$ and the total number of employed persons $\text{RATIO}_{\text{total}}$. An ideal ratio of 1 indicates that all components of the summation correctly add up to the respective subtotal or total.

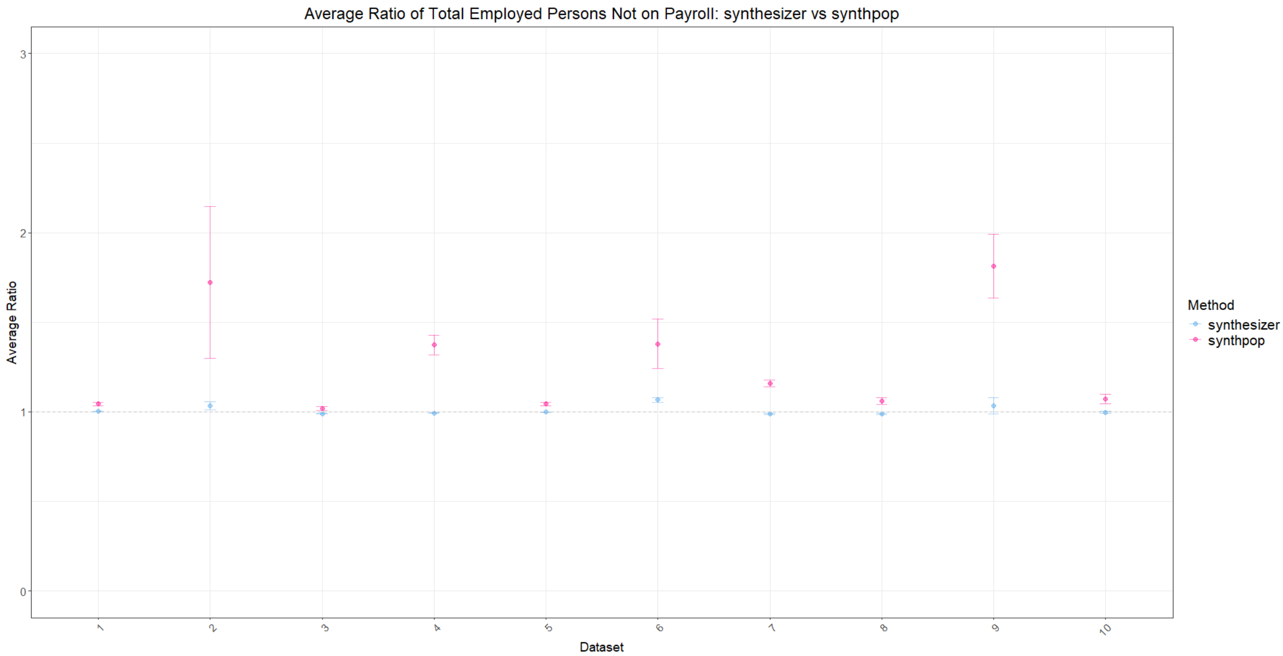


Figure 18: $\text{RATIO}_{\text{not on payroll}}$ results (averaged over 15 synthetic datasets) for real data setting. Synthetic data generation (SDG) methods are represented in different colours. The grey dashed line represents the line where $y = 1$. Points that are closer to this line are preferred, as this line indicates the point where the additive relationship is perfectly upheld.

The results for the average ratios of employed persons not on the payroll are shown in Figure 18, with the datasets on the x-axis and the average ratio (averaged over 15 synthetic datasets) on the y-axis. The dashed line represents the line where $y = 1$, and since the ratios should equal one, points closer to the line are preferred. The findings reveal that `synthesizer` achieves ratios very close to 1 for all datasets. Further, `synthesizer` outperforms `synthpop` clearly for all datasets, having ratios closer to 1. The superior performance of `synthesizer` is likely due to its rank-matching approach, which effectively preserves additive relationships.

An interesting observation is the large ratios for `synthpop` in datasets 2 and 9 particularly, where the ratios are rather large. This suggests that `synthpop`'s iterative procedure may have overcomplicated the relationships, leading to its inability to uphold the additive structure accurately.

To further assess the preservation of additive relationships, the ratios for the total employed persons are analyzed. The results are presented in Figure 19. The datasets are shown on the x-axis and the average ratio on the y-axis. The dashed line represents the line where $y = 1$, and since the ratios should equal one, points closer to the line are preferred.

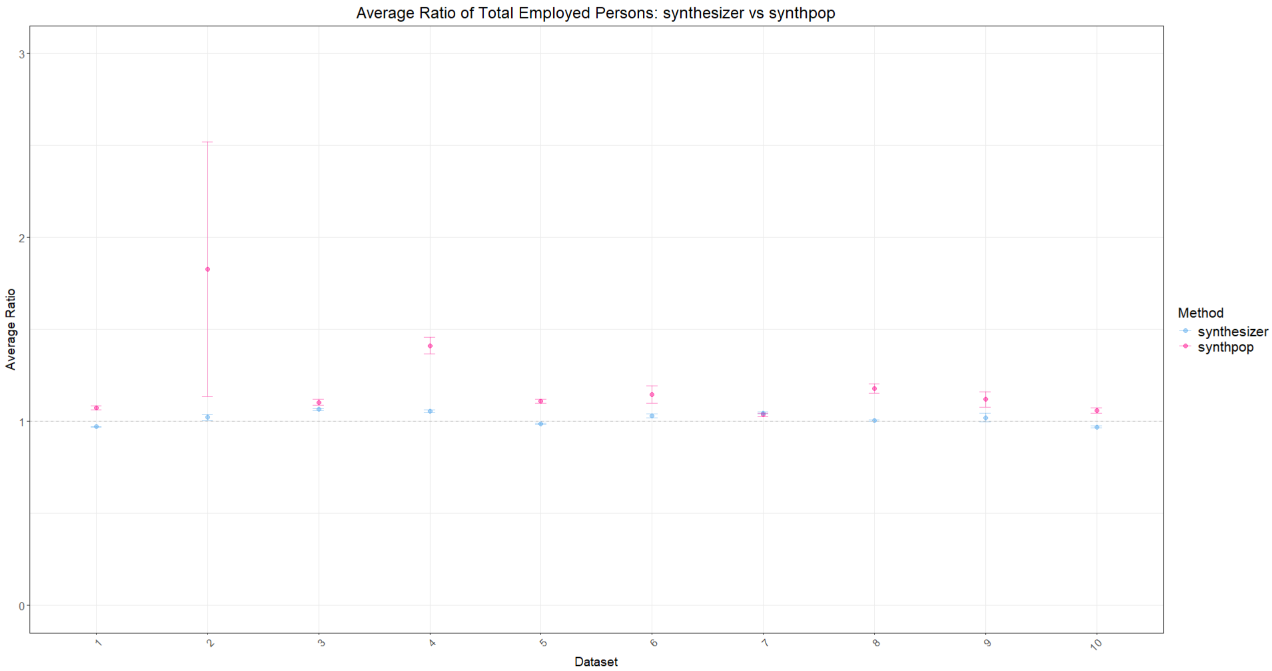


Figure 19: $\text{RATIO}_{\text{total}}$ results for real datasets. Synthetic data generation (SDG) methods are represented in different colours. The grey dashed line represents the line where $y = 1$. Points that are closer to this line are preferred, as this line indicates the point where the additive relationship is perfectly upheld.

For this analysis, the points for `synthesizer` showed more deviation from the line $y = 1$ compared to the previous ratio. Recall that total employed persons not on payroll is a subtotal of total employed persons. The additional variables added to total employed persons not on payroll (because total employed persons = total employed persons not on payroll + total employed persons on payroll) seems to have distorted the ratios for `synthesizer`. This suggests that while the rank-matching implemented in `synthesizer` does well for upholding the mathematical restrictions associated with subtotals, it does slightly worse for totals. However, more of these ratios would need to be analysed for this to be confirmed.

However, despite this, and with the exception of dataset 7, `synthesizer` still outperforms `synthpop`. More specifically, the outlier for `synthpop` in dataset 2 persists, with an inflated ratio and wide CI. This suggests that the iterative procedure employed by `synthpop` still struggles to uphold the additive structure for this specific dataset, but the reason why it is struggling with this particular dataset is unclear. It could be because it has a very large ratio of records to variables, and because `synthpop` learns from all variables, it is getting confused about the relationships between variables.

These findings underscore the robustness of `synthesizer` in preserving additive relationships, even as `synthpop` exhibits occasional challenges, and this is likely due to the rank-matching approach implemented by `synthesizer`.

Chapter 6

Discussion

This thesis provides a comprehensive evaluation of the `synthesizer` R package for synthetic data generation, focusing on its ability to replicate key characteristics of various datasets. In this chapter, results and conclusions will be discussed with respect to the three research questions outlined in chapter 1.

6.1 Evaluating Synthetic Data Quality (RQ 1)

This thesis project involved a thorough assessment of the literature on synthetic data quality. Literature showed that the quality of synthetic data is multi-faceted and should ideally be evaluated with respect to the intended application of the synthetic data. This thesis project used some metrics popular in the literature to assess the fidelity and specific utility. Further, some additional metrics were implemented to assess whether the `synthesizer` package replicates the properties common in economic and financial datasets.

Five metrics were selected, namely the propensity mean squared error (pMSE), a comparison of percentage of zero values, a comparison of percentage of missing values, a comparison of F_1 scores and ratios of the components of a total to that total.

While this study focused on these metrics to get a preliminary assessment of the distributional similarity between the synthetic and real data, and to see if the method replicates the attributes of economic datasets, future evaluations should expand to include broader metrics for distributional similarity and utility. Further, privacy protection assessments, crucial for real-world applications, remain unexplored in this work, and should be evaluated prior to the release of synthetic data.

6.2 Evaluating the `synthesizer` method against the `synthpop` method using the selected metrics (RQ 2 & RQ 3)

Using a systematic approach, the performance of `synthesizer` was evaluated on three data types: multivariate normal (MVN) datasets, zero-inflated log-normal datasets, and real-world economic and financial datasets. The findings underscore the strengths and limitations of `synthesizer` in comparison to the widely used non-parametric `synthpop` method for synthetic data generation across different scenarios. The findings will be discussed with respect to each metric.

6.2.1 Propensity mean squared error (pMSE)

For the multivariate normal (MVN) simulations, the performance of `synthesizer` was consistent with `synthpop` when there was more overlap between the groups in the data. However, the performance of `synthesizer` declined (pMSE increased) when the means of the two groups in data were further apart, indicating the method's limitations in capturing distributions without dependencies. Further, the performance of `synthesizer` deteriorated in scenarios with correlations of 0.75, as reflected by higher pMSE values. Conversely, `synthpop` performed better with more correlated variables and struggled with less correlated variables. This suggests that `synthesizer` works better with moderately correlated variables, but that rank matching becomes less effective when correlation exceeds 0.75 compared to `synthpop`, where `synthpop`'s iterative learning excels.

In zero-inflated log-normal simulations, `synthpop` consistently outperformed `synthesizer` in terms of pMSE (smaller values). This is attributed to the highly correlated nature of the variables since they were simulated such that each added variable consisted of the previous variable plus some noise. Therefore, each variable was related to the variable before it. This favoured the way in which `synthpop` simulates data, as `synthpop` iteratively uses one variable and combinations of variables to predict and learn patterns in other variables.

In the real data, `synthpop` outperformed `synthesizer` with lower pMSE values for larger ratios of records to variables, suggesting that the method is able to better utilise a greater ratio of records to variables to learn the relationships in the data compared to `synthesizer`. For both simulations and the real data, both methods observed an increase in pMSE, and therefore a decline in performance with a decreasing ratio of records to variables. This emphasized that a sufficient number of records is needed for each variable to learn the relationships in the data. Although, no exact optimal ratio of records to variables was found, performance in both methods had a more clear decline when the ratio of records to variables was less than 8:1.

6.2.2 Percentage zero values

The analysis of simulated zero-inflated log-normal datasets and real datasets demonstrated that both methods performed well in maintaining consistent patterns of zero values as per the real data. While both methods performed well, `synthesizer` exhibited slightly neater linear relationships with the percentage of zero values, and as a result had lower MSE scores in most settings.

This is because the `synthesizer` algorithms samples zero values for a variable in proportion to their occurrence in the real data. Whereas `synthpop` does not explicitly sample zero values and instead learns relationships in the data for one variable from other variables.

In few cases, synthpop did have lower MSE scores than synthesizer. A possible explanation may be that in these cases, the presence of zero values for a record might be correlated between variables, allowing synthpop to effectively replicate their occurrence in the synthetic data. However, another possibility is that this happened randomly due to a lack of iterations.

6.2.3 Percentage missing values

The real data analysis highlighted both methods' ability to replicate missing values well. While both performed well, it was expected that the results would be similar to those of the zero percentages, in that synthesizer would would outperform synthpop, as it deals with missing values in a similar fashion to zero values.

It was interesting to observe that synthpop struggled more with replicating missing values than expected. A possible explanation could be that while the presence of zero values for a record might be correlated between variables, it might be that the presence of missing values is not as correlated between variables as the presence of zero values, and, therefore, synthpop struggles to learn the relationship between the appearance of missing values in the data.

Overall, explicitly sampling missing values in proportion to their occurrence in the real data for each variable, as done in synthesizer worked well for replicating missing values in synthetic data.

6.2.4 F_1 score

In terms of serving as training data for prediction tasks, synthesizer showed promise. The main finding here was that slightly lower precision in replicating real data (higher pMSE) often translated to higher F_1 scores (sometimes even higher than those of the real data), likely due to reduced over-fitting in comparison to the original data. However, when the pMSE was too high, the synthetic data lost its utility as training data and F_1 scores dropped since the synthetic data was too different from the real test data to effectively train a model.

6.2.5 Ratios

synthesizer did well in replicating additive relationships (most ratios were close to 1). While synthpop also generally performed well, it occasionally exhibited some outliers in additive relationships, likely due to challenges in its iterative synthesis approach. The results suggest that synthesizer's rank matching procedure works well for preserving additive relationships. However, these relationships are still not perfectly preserved for either method.

6.2.6 Conclusion

While there is no clear optimal package, the results demonstrate `synthesizer`'s strengths in mimicking properties of economic datasets, and its intended applications. Specifically, `synthesizer` replicates zero inflation, missing values, and additive relationships well, often more effectively than `synthpop`. Moreover, `synthesizer` offers a simpler, faster method, making it less computationally intensive.

However, for settings when the prediction problem offers clear separation, `synthesizer` does not work well, and `synthpop` should be used. This is because the CART model implemented in `synthpop` can easily identify this separation at the first split. Whereas, for `synthesizer`, which samples from the inverse eCDF, this is challenging. `synthesizer` might perform better if the two halves of the data were synthesized separately and then merged. `synthpop` also outperformed `synthesizer` in settings where variables are highly correlated, because of its ability to iteratively use previous variables to predict a variable in the data. This proved more effective than the rank-matching implemented by `synthesizer` in situations of very high correlations.

The influence of dataset size was key across all three data types. Both `synthesizer` and `synthpop` exhibited a decline in performance as the ratio of records to variables decreased in the simulated and real datasets, emphasizing the importance of adequate numbers of records relative to the number of variables. While there is no defined optimal ratio, this study shows the influence of dataset size plays a key role in the quality of data synthesized.

6.2.7 Limitations and Further Research

Though this study used three different dataset types to evaluate the method, the simulated datasets were kept relatively simple. The multivariate normal (MVN) datasets were generated with uniform correlations between variables, limiting the exploration of varying correlation structures. Further, only 4 correlation values were implemented. Future studies could look more at different correlation values, particularly at low and high values. This can be done with a simple simulation with few variables, where pMSE is evaluated when correlation values are very low, to see for what values `synthesizer` starts performing better, and for higher values, to see when the rank-matching in `synthesizer` starts to struggle. Also, future experiments could consider varying the correlations between variables, so that not all variables are equally correlated.

In the log-normal simulations, the experiments were constrained to a single zero percentage of 0.1, leaving the impact of varying zero-inflation levels unexplored. Further, in the real datasets, there may be a correlation between the probability of two variables having a zero value, which was also not simulated. Further research could also examine the impact of diverse zero-inflation levels, by possible introducing different zero percentages into the data, or correlating the way in which zero values are introduced. This would identify whether `synthesizer` does just as well at preserving zero values in different contexts.

Further, simulated datasets could be updated to more closely mimic the real datasets, to get a more detailed analysis of the synthesizer method in a more realistic context. For example, relationships common to economic datasets such as $X + Y = Z$ could be introduced into the simulated data, or missing values could be introduced.

Data synthesis was limited to 15 replications due to synthpop's computational intensity. While this was sufficient for point estimates, this limited the convergence of the distributions of these point estimates. Future studies could explore more optimal replication counts by seeing how many replications it takes for the distributions of the point estimates of the quality metrics to converge. However, this may be computationally costly, particularly for synthpop, and the trade-off between computational efficiency and synthesis quality for synthesizer and synthpop should be explored in more detail.

6.3 Conclusion

By identifying these strengths and limitations, this study lays the groundwork for refining the synthesizer package. However, further research is necessary to get more exact rules for when the method fails to work, and to see if the method preserves the privacy of the entities represented in the data before any synthetic data can be released.

Acknowledgements

I would like to express my gratitude to my thesis supervisors Dr. Mark van der Loo and Dr. Sanne Willems for their continued encouragement, support and guidance throughout this project.

To Mark, I am thankful for our weekly meetings, during which you shared valuable insights that significantly enhanced my understanding of not only synthetic data, but many other statistical concepts and methodologies.

To Sanne, I greatly appreciate the time and effort you dedicated to reviewing my drafts and providing extensive feedback. Your detailed comments have been crucial in improving the clarity and depth of my work.

I would also like to extend my heartfelt thanks to CBS for providing me with the opportunity to conduct my thesis within their organization. The resources and support offered created an enriching and inspiring working environment.

Bibliography

[1] J. Drechsler and A.-C. Haensch, "30 years of synthetic data," *arXiv preprint arXiv:2304.02107*, 2023. [Online]. Available: `http://arxiv.org/abs/2304.02107`.

[2] UNECE, "Synthetic data for official statistics," United Nations, Tech. Rep., 2022.

[3] CBS. "Cbs multi-annual programme 2024-2028." (2024), [Online]. Available: `https://www.cbs.nl/en-gb/longread/diversen/2022/cbs-multi-annual-programme-2024-2028` (visited on 09/16/2024).

[4] The Parliament of Netherlands, *Statistics netherlands act*, 2022. [Online]. Available: `cbs.nl/en-gb/about-us/who-we-are/our-course/overview-of-multi-annual-programmes-and-annual-reports/statistics-netherlands-act`.

[5] J. Jordon, L. Szpruch, F. Houssiau, *et al.*, "Synthetic data-what, why and how?" The Alan Turing Institute, Tech. Rep., 2022.

[6] M. Sluiskes, "Imputation of business survey data: A systematic comparison between ratio and random forest-based imputation methods," M.S. thesis, Universiteit Leiden, 2021.

[7] M. van der Loo. "Synthesizer." (2024), [Online]. Available: `https://CRAN.R-project.org/package=synthesizer` (visited on 09/16/2024).

[8] F. K. Dankar, M. K. Ibrahim, and L. Ismail, "A multi-dimensional evaluation of synthetic data generators," *IEEE Access*, vol. 10, pp. 11 147–11 158, 2022.

[9] V. Bernardo, "Synthetic data," European Data Protection Supervisor (EDPS), Tech. Rep., 2021.

[10] D. B. Rubin, "Statistical disclosure limitation," *Journal of Official Statistics*, 1993.

[11] J. Drechsler, S. Bender, and S. Rässler, "Comparing fully and partially synthetic datasets for statistical disclosure control in the german iab establishment panel," *Transactions on Data Privacy*, vol. 1, pp. 105–130, Dec. 2008.

[12] McKinsey & Company. "What is deep learning?" (2024), [Online]. Available: https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-deep-learning (visited on 11/12/2024).

[13] D. P. Kingma and M. Welling, *Auto-encoding variational bayes*, 2022. arXiv: 1312.6114 [stat.ML]. [Online]. Available: https://arxiv.org/abs/1312.6114.

[14] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, *Generative adversarial networks*, 2014. arXiv: 1406.2661 [stat.ML]. [Online]. Available: https://arxiv.org/abs/1406.2661.

[15] O. S. "Introduction to sampling methods." (2023), [Online]. Available: https://towardsdatascience.com/introduction-to-sampling-methods-c934b64b6b08 (visited on 10/10/2024).

[16] K. H. Torsten Hothorn and A. Zeileis, "Unbiased recursive partitioning: A conditional inference framework," *Journal of Computational and Graphical Statistics*, vol. 15, no. 3, pp. 651–674, 2006. DOI: 10.1198/106186006X133933. [Online]. Available: https://doi.org/10.1198/106186006X133933.

[17] T. Volker, *Practical workshop on creating synthetic data*, https://thomvolker.github.io/osf_synthetic/, Accessed: 2024-12-11, 2024.

[18] V. S. Chundawat, A. K. Tarun, M. Mandal, M. Lahoti, and P. Narang, "A universal metric for robust evaluation of synthetic tabular data," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 1, pp. 300–309, 2024. DOI: 10.1109/TAI.2022.3229289.

[19] J. Snoke, G. Raab, B. Nowok, C. Dibben, and A. Slavkovic, "General and specific utility measures for synthetic data," *arXiv preprint arXiv:1604.06651*, 2016. [Online]. Available: http://arxiv.org/abs/1604.06651.

[20] C. Arnold and M. Neunhoeffer, *Really useful synthetic data – a framework to evaluate the quality of differentially private synthetic data*, 2021. arXiv: 2004.07740 [stat.ML]. [Online]. Available: https://arxiv.org/abs/2004.07740.

[21] F. J. Sarmin, A. R. Sarkar, Y. Wang, and N. Mohammed, *Synthetic data: Revisiting the privacy-utility trade-off*, 2024. arXiv: 2407.07926 [cs.CR]. [Online]. Available: https://arxiv.org/abs/2407.07926.

[22] J. Achterberg, M. Haas, and M. Spruit, "On the evaluation of synthetic longitudinal electronic health records," *BMC Medical Research Methodology*, Apr. 2024. DOI: 10.1186/s12874-024-02304-4.

[23] A. Hundepool, J. Domingo-Ferrer, L. Franconi, *et al.*, *Statistical disclosure control*. John Wiley & Sons, 2012.

[24] J. Domingo-Ferrer, J. Mateo-Sanz, and V. Torra, "Comparing sdc methods for microdata on the basis of information loss and disclosure," *Proceedings of ETK-NTTS 2001*, Jan. 2001.

[25] J. C. Taub, "Synthetic data: An exploration of data utility and disclosure risk," Ph.D. dissertation, The University of Manchester, 2020.

[26] M.-J. Woo, J. P. Reiter, A. Oganian, and A. F. Karr, "Global measures of data utility for microdata masked for disclosure limitation," *Journal of Privacy and Confidentiality*, vol. 1, no. 1, Apr. 2009. DOI: 10.29012/jpc.v1i1.568. [Online]. Available: https://journalprivacyconfidentiality.org/index.php/jpc/article/view/568.

[27] J. Snoke and A. Slavković, *Pmse mechanism: Differentially private synthetic data with maximal distributional similarity*, 2018. arXiv: 1805.09392 [stat.ME]. [Online]. Available: https://arxiv.org/abs/1805.09392.

[28] K. El Emam, L. Mosquera, X. Fang, and A. El-Hussuna, "Utility metrics for evaluating synthetic health data generation methods: Validation study," *JMIR Medical Informatics*, vol. 10, no. 4, Apr. 2022, ISSN: 2291-9694. DOI: 10.2196/35734. [Online]. Available: https://medinform.jmir.org/2022/4/e35734.

[29] D. Gunay. "Cart." (2023), [Online]. Available: https://medium.com/@denizgunay/cart-4838fec3e405 (visited on 11/20/2024).

[30] P. Patil. "What is exploratory data analysis?" (2018), [Online]. Available: https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15 (visited on 11/20/2024).

[31] Z. Zhao, A. Kunar, H. V. der Scheer, R. Birke, and L. Y. Chen, *Ctab-gan: Effective table data synthesizing*, 2021. arXiv: 2102.08369 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2102.08369.

[32] Y. M. J. Woo and A. Slavkovic, "Generalised linear models with variables subject to post randomization method," *Statistica Applicata-Italian Journal of Applied Statistics*, vol. 24, no. 1, pp. 29–56, 2015.

[33] D. Rankin, M. Black, R. Bond, J. Wallace, M. Mulvenna, and G. Epelde, *Reliability of supervised machine learning when using synthetic healthcare data: A model to preserve privacy for data sharing (preprint)*, Mar. 2020. DOI: `10.2196/preprints.18910`.

[34] M. Hittmeir, A. Ekelhart, and R. Mayer, "On the utility of synthetic data: An empirical evaluation on machine learning tasks," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, ser. ARES '19, Canterbury, CA, United Kingdom: Association for Computing Machinery, 2019, ISBN: 9781450371643. DOI: `10.1145/3339252.3339281`. [Online]. Available: `https://doi.org/10.1145/3339252.3339281`.

[35] C. Esteban, S. L. Hyland, and G. Rätsch, *Real-valued (medical) time series generation with recurrent conditional gans*, 2017. arXiv: `1706.02633 [stat.ML]`. [Online]. Available: `https://arxiv.org/abs/1706.02633`.

[36] A. Aysha. "Evaluate synthetic data quality using downstream ml." (2023), [Online]. Available: `https://mostly.ai/blog/synthetic-data-quality-evaluation` (visited on 11/20/2024).

[37] IBM. "What is logistic regression?" (2024), [Online]. Available: `https://www.ibm.com/topics/logistic-regression` (visited on 11/21/2024).

[38] A. Vina. "What is f1 score? a computer vision guide." (2024), [Online]. Available: `https://blog.roboflow.com/f1-score/` (visited on 11/21/2024).

[39] dremio. "One-vs-all classification." (2024), [Online]. Available: `https://www.dremio.com/wiki/one-vs-all-classification/` (visited on 01/10/2025).

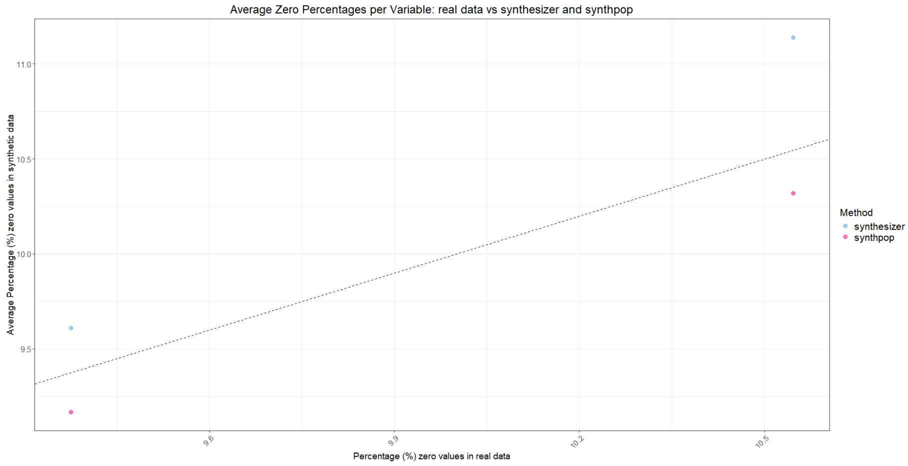
Appendix A

Percentage Zero Value Plots

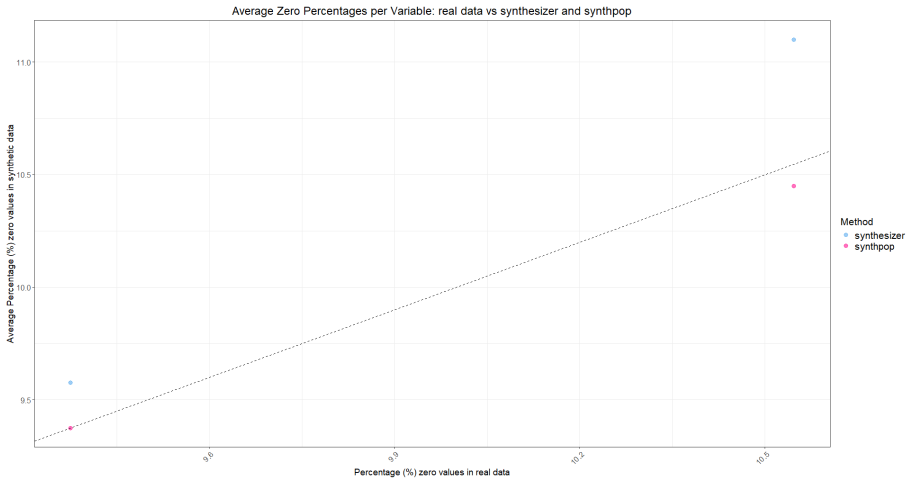
This appendix presents the plots of the results for the percentage of zero values in the synthesized datasets for all the zero-inflated log-normal simulated datasets and all the real datasets.

A.1 Zero-Inflated Log-normal Data

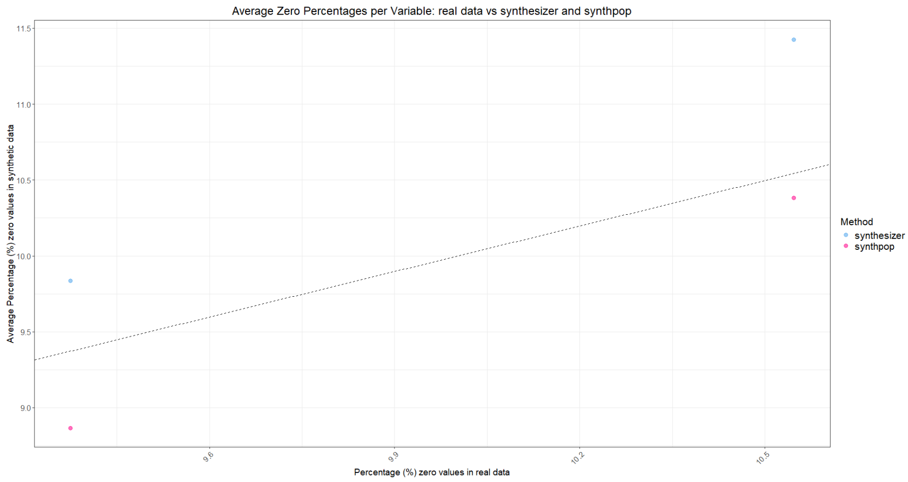
The following pages contain the plots for the average percentages of zero values in the synthetic datasets plotted against the percentage of zero values in the simulated data for the zero-inflated log-normal simulation.



(a) Dataset with 2 variables and $\mu_2 = 2.25$

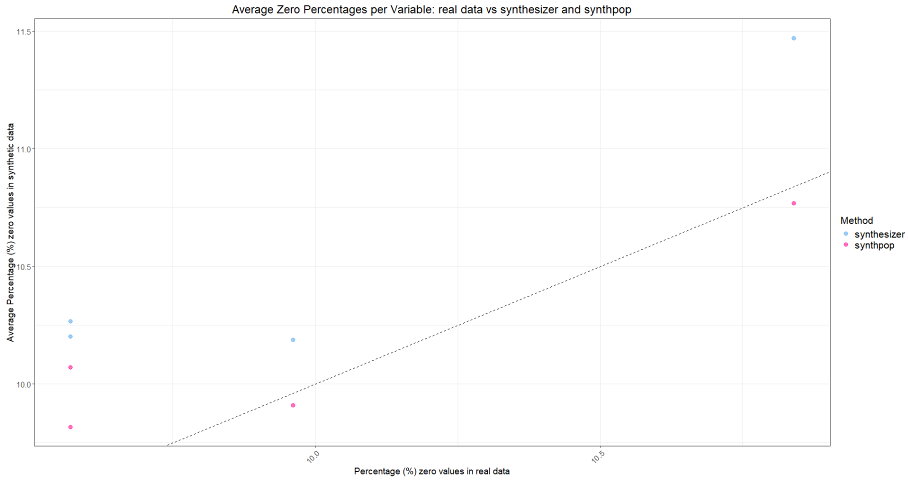


(b) Dataset with 2 variables and $\mu_2 = 3.5$

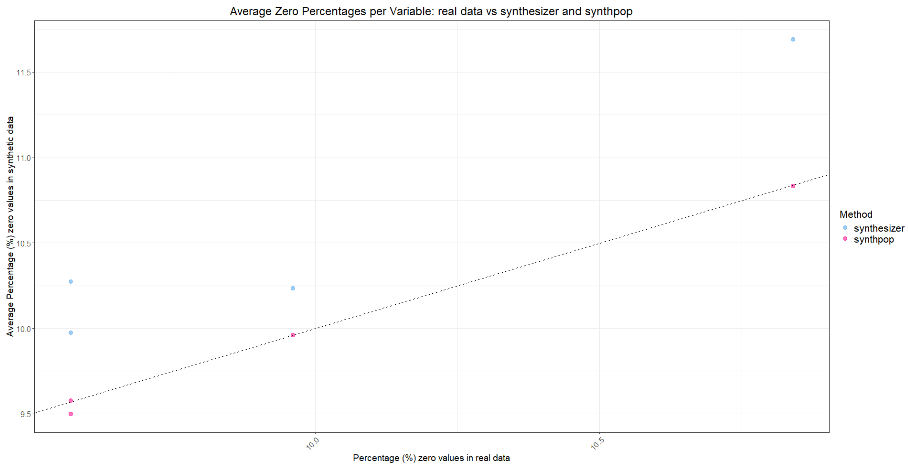


(c) Dataset with 2 variables and $\mu_2 = 6$

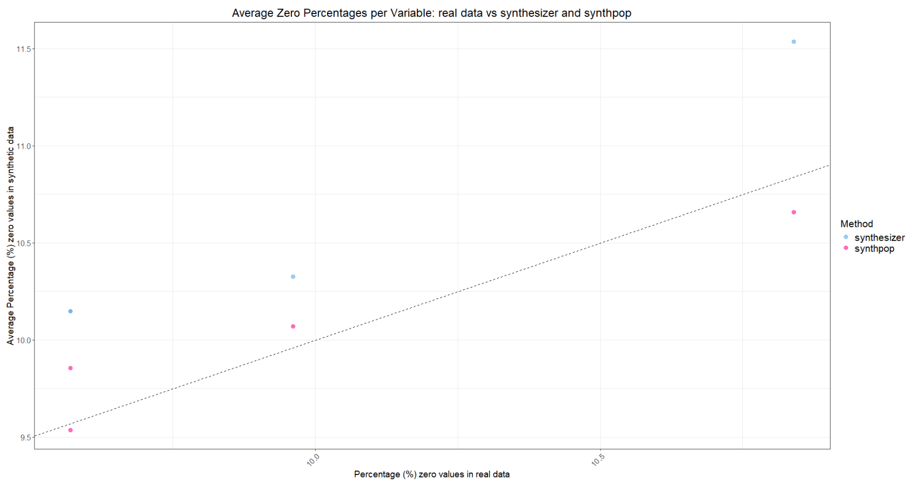
Figure 20: Plots showing the average percentage of zero values in synthetic datasets compared to the percentage of zero values in the simulated data for datasets containing 2 variables



(a) Dataset with 4 variables and $\mu_2 = 2.25$

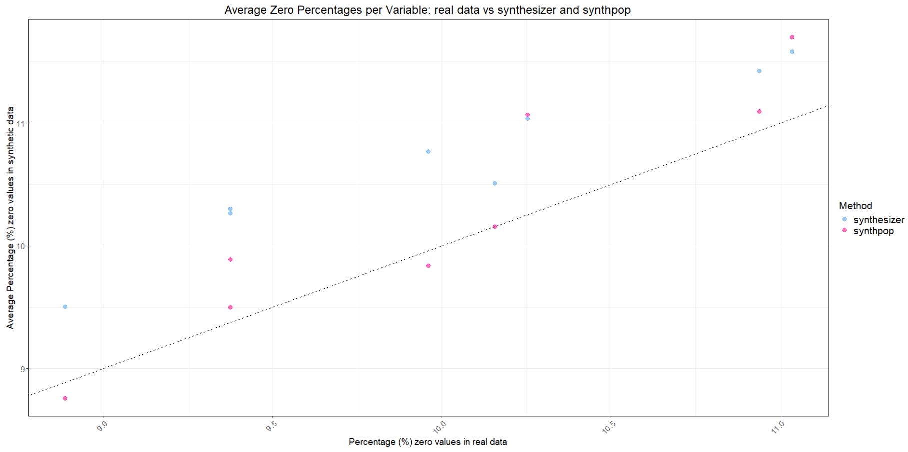


(b) Dataset with 4 variables and $\mu_2 = 3.5$

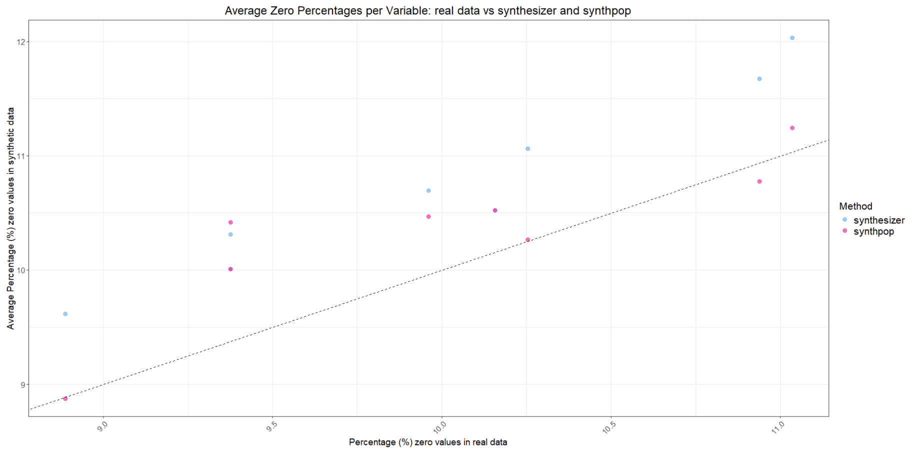


(c) Dataset with 4 variables and $\mu_2 = 6$

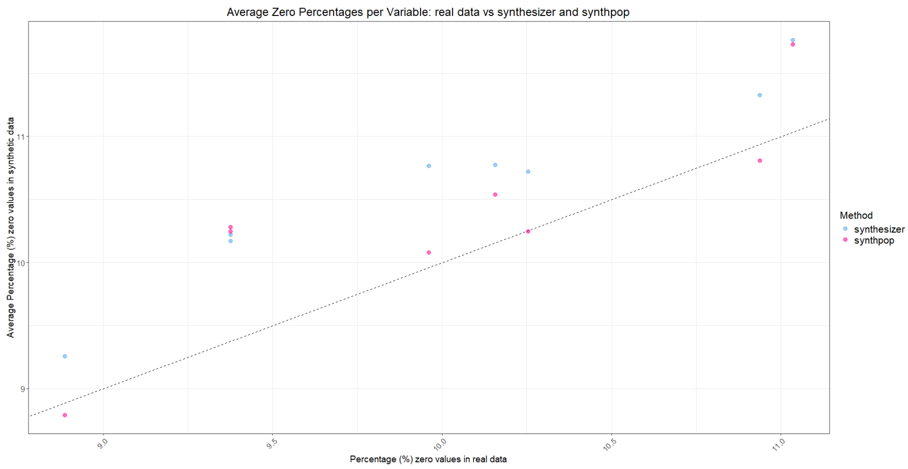
Figure 21: Plots showing the average percentage of zero values in synthetic datasets compared to the percentage of zero values in the simulated data for datasets containing 4 variables



(a) Dataset with 8 variables and $\mu_2 = 2.25$

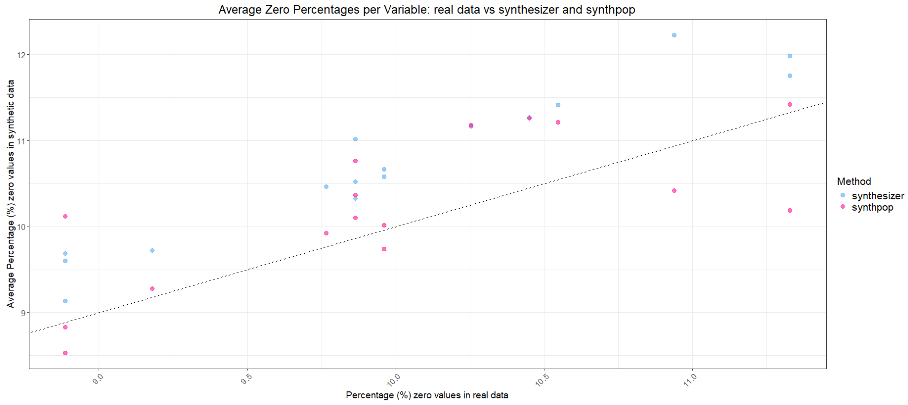


(b) Dataset with 8 variables and $\mu_2 = 3.5$

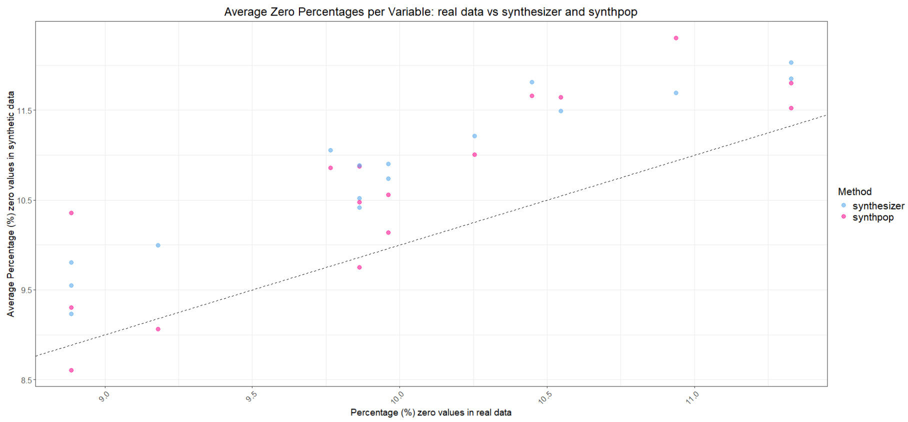


(c) Dataset with 8 variables and $\mu_2 = 6$

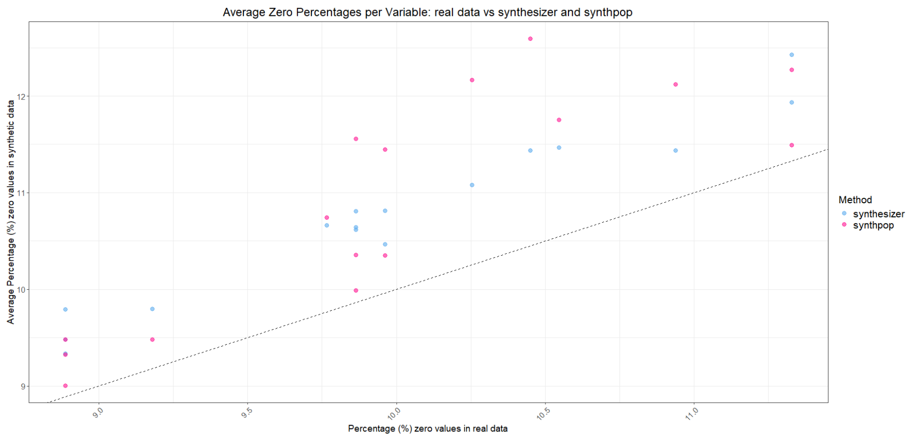
Figure 22: Plots showing the average percentage of zero values in synthetic datasets compared to the percentage of zero values in the simulated data for datasets containing 8 variables



(a) Dataset with 16 variables and $\mu_2 = 2.25$

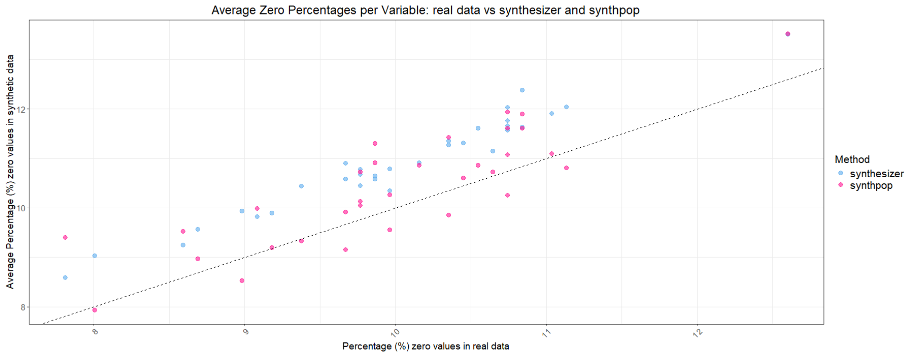


(b) Dataset with 16 variables and $\mu_2 = 3.5$

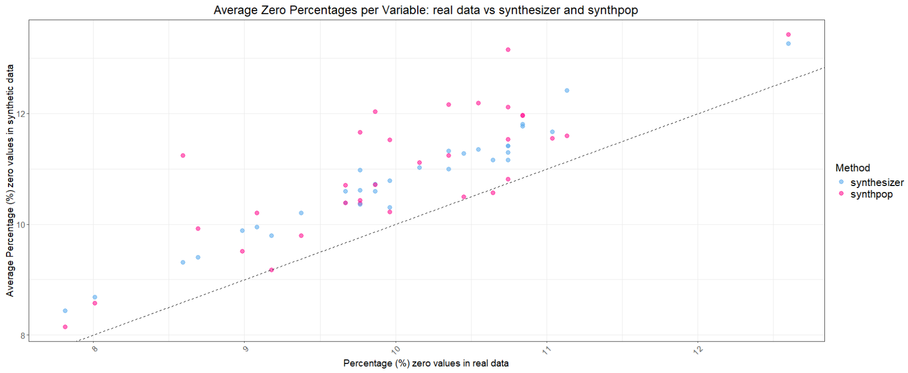


(c) Dataset with 16 variables and $\mu_2 = 6$

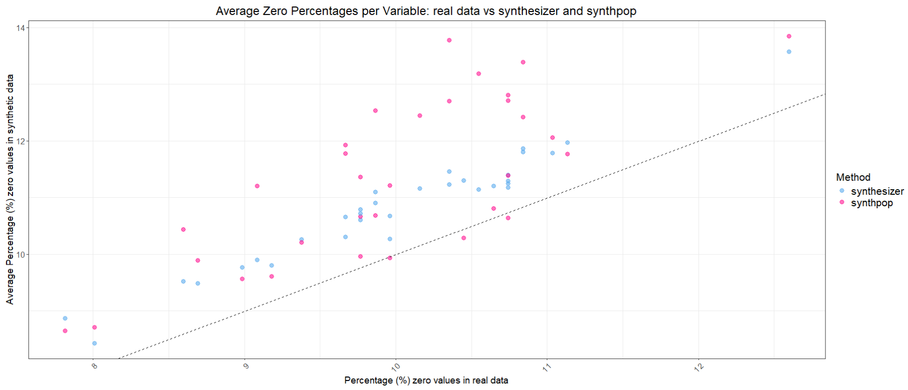
Figure 23: Plots showing the average percentage of zero values in synthetic datasets compared to the percentage of zero values in the simulated data for datasets containing 16 variables



(a) Dataset with 32 variables and $\mu_2 = 2.25$

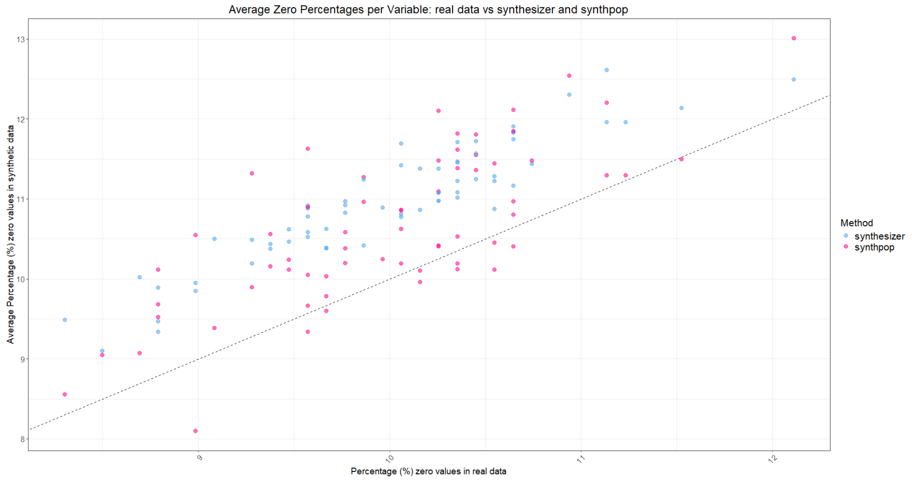


(b) Dataset with 32 variables and $\mu_2 = 3.5$

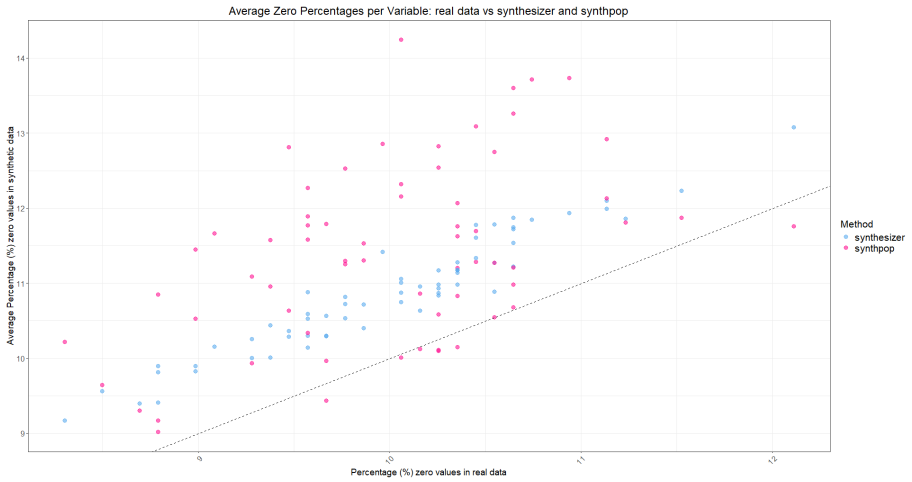


(c) Dataset with 32 variables and $\mu_2 = 6$

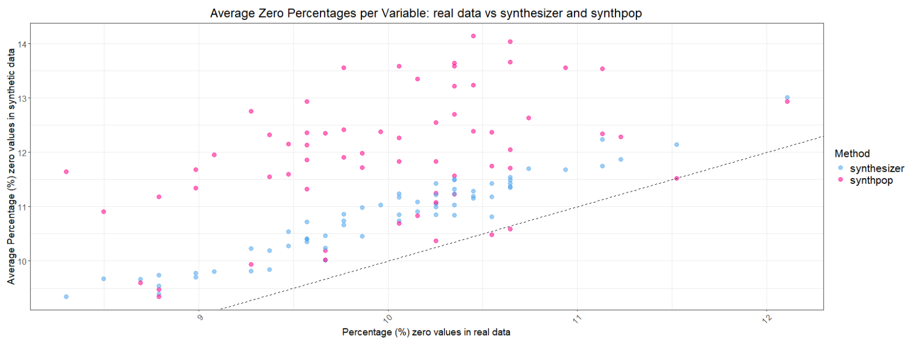
Figure 24: Plots showing the average percentage of zero values in synthetic datasets compared to the percentage of zero values in the simulated data for datasets containing 32 variables



(a) Dataset with 64 variables and $\mu_2 = 2.25$

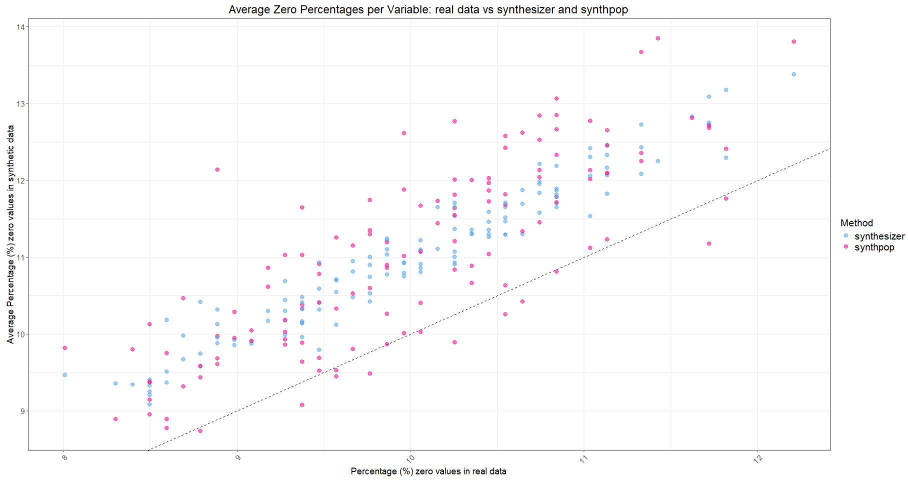


(b) Dataset with 64 variables and $\mu_2 = 3.5$

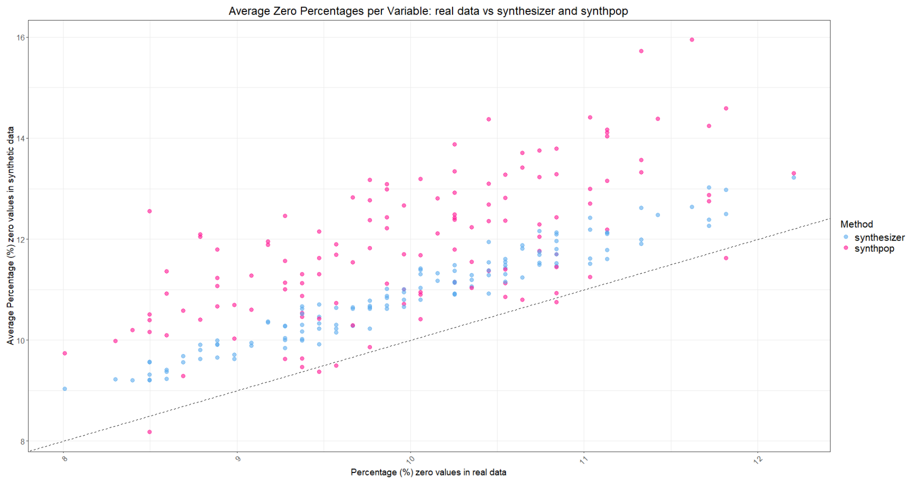


(c) Dataset with 64 variables and $\mu_2 = 6$

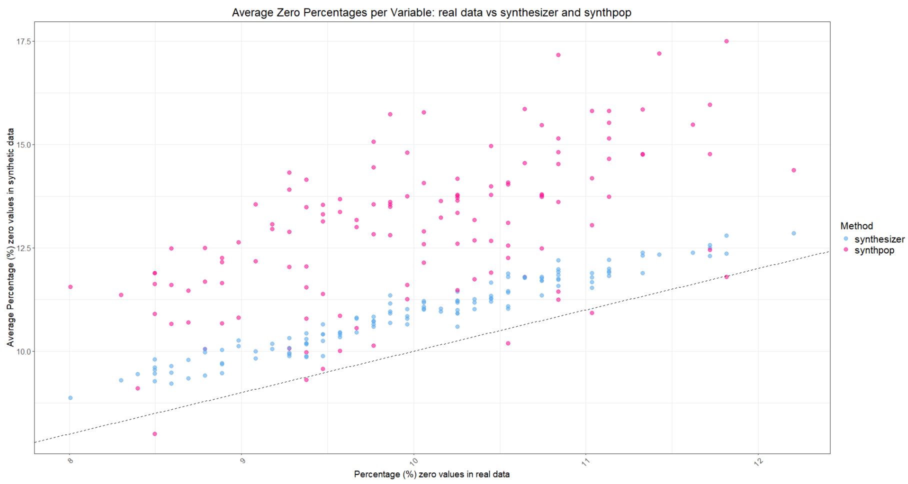
Figure 25: Plots showing the average percentage of zero values in synthetic datasets compared to the percentage of zero values in the simulated data for datasets containing 64 variables



(a) Dataset with 128 variables and $\mu_2 = 2.25$

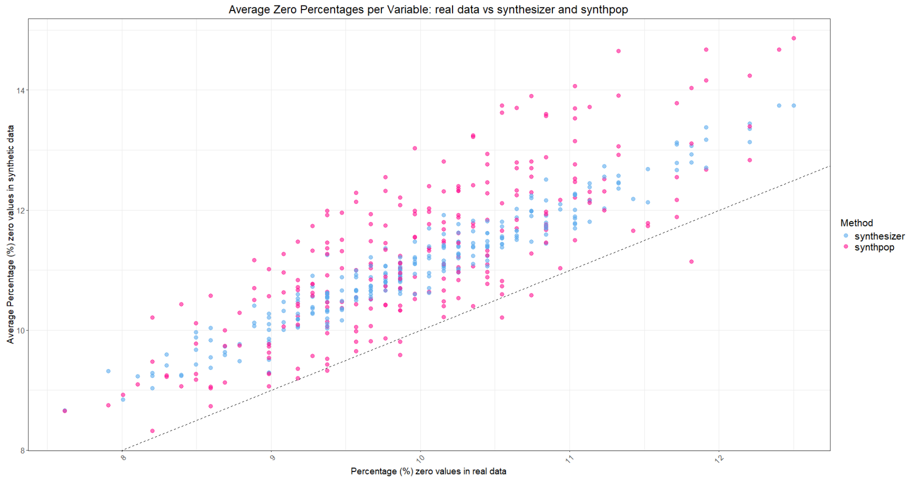


(b) Dataset with 128 variables and $\mu_2 = 3.5$

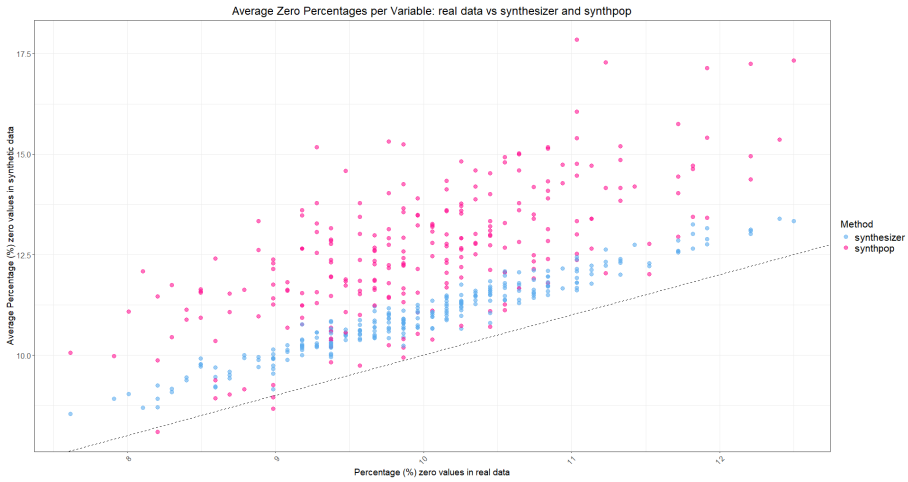


(c) Dataset with 128 variables and $\mu_2 = 6$

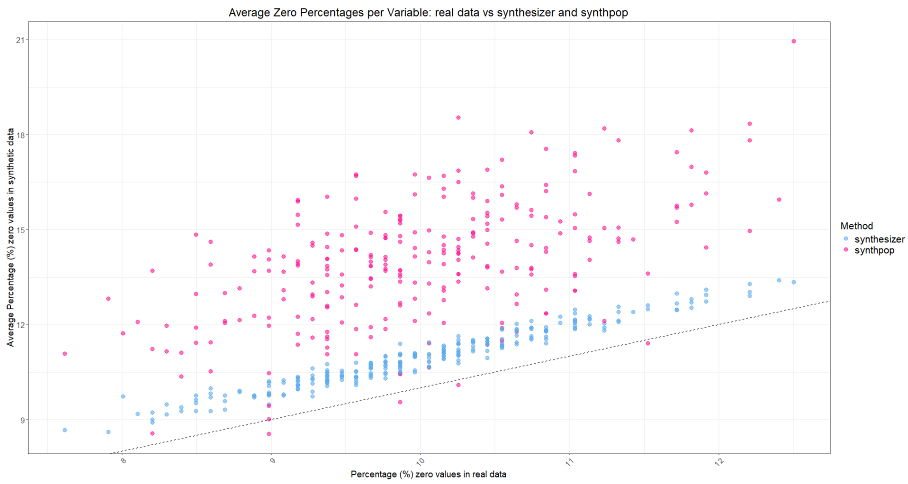
Figure 26: Plots showing the average percentage of zero values in synthetic datasets compared to the percentage of zero values in the simulated data for datasets containing 128 variables



(a) Dataset with 256 variables and $\mu_2 = 2.25$

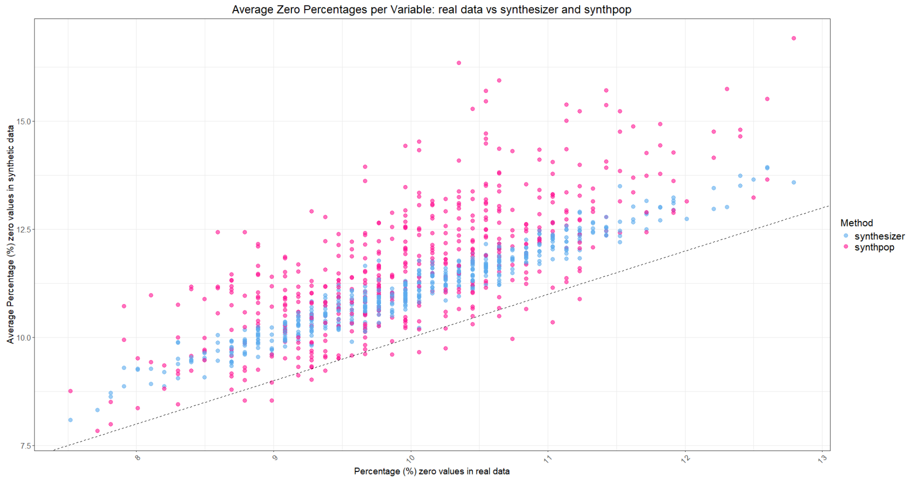


(b) Dataset with 256 variables and $\mu_2 = 3.5$

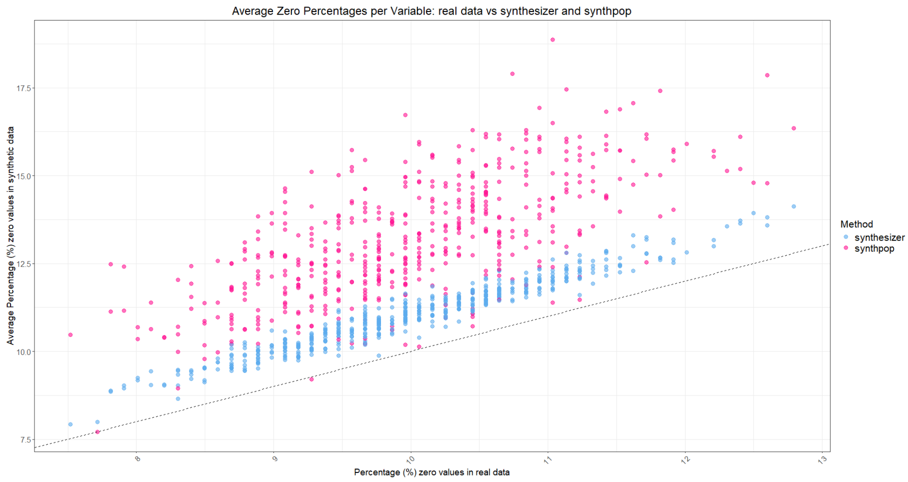


(c) Dataset with 256 variables and $\mu_2 = 6$

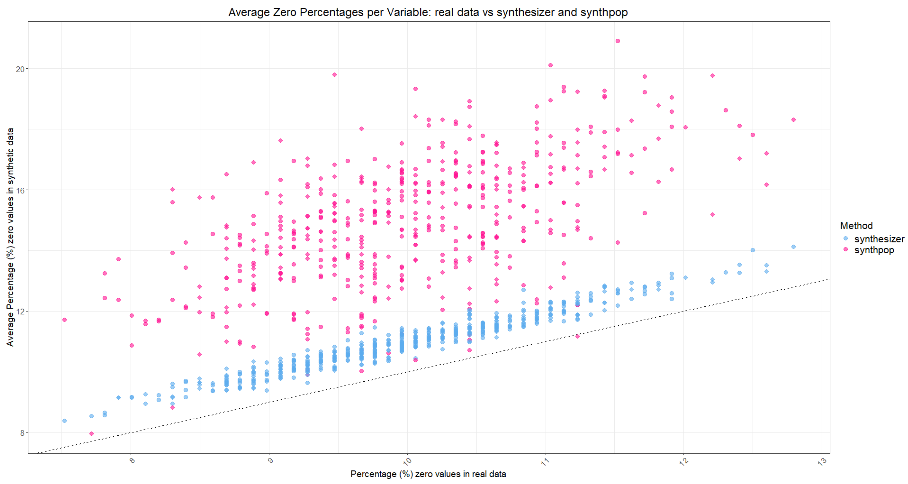
Figure 27: Plots showing the average percentage of zero values in synthetic datasets compared to the percentage of zero values in the simulated data for datasets containing 256 variables



(a) Dataset with 512 variables and $\mu_2 = 2.25$



(b) Dataset with 512 variables and $\mu_2 = 3.5$

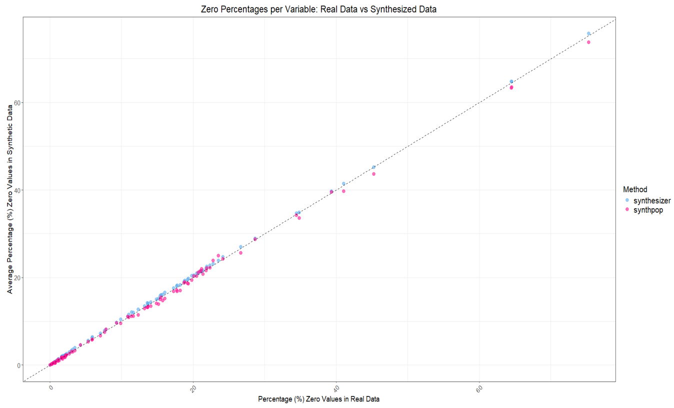


(c) Dataset with 512 variables and $\mu_2 = 6$

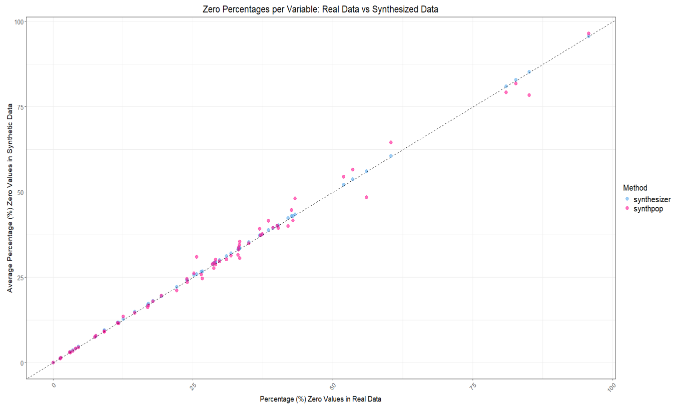
Figure 28: Plots showing the average percentage of zero values in synthetic datasets compared to the percentage of zero values in the simulated data for datasets containing 512 variables

A.2 Real Data

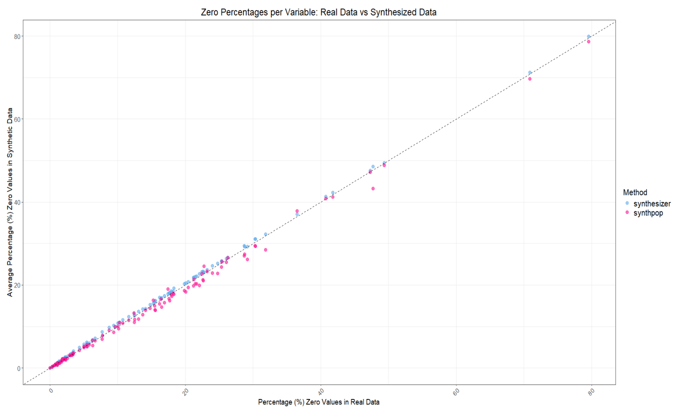
The following pages contain the plots for the average percentages of zero values in the synthetic datasets plotted against the percentage of zero values for the 10 real datasets.



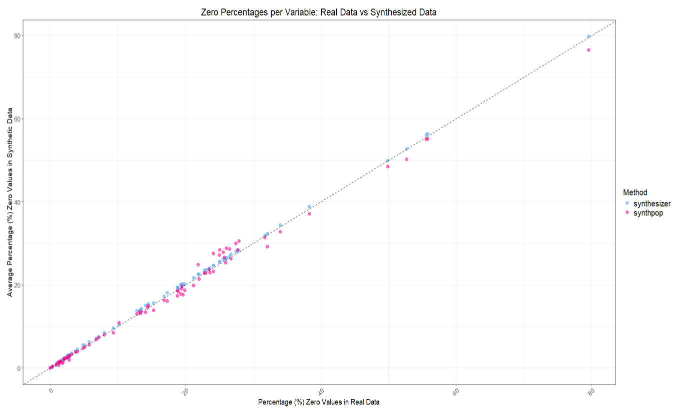
(a) Dataset 1



(b) Dataset 2

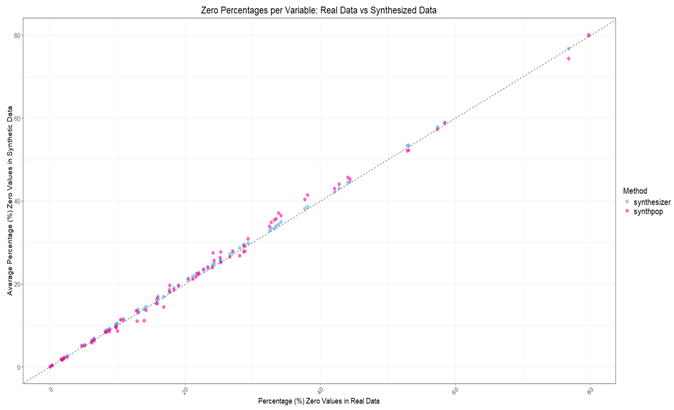


(c) Dataset 3



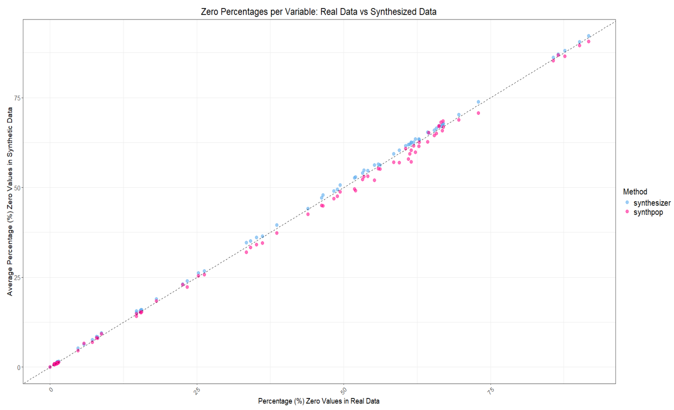
(d) Dataset 4

Figure 29: Zero Percentages in Real and Synthesized Datasets using both `synthesizer` and `synthpop` for real datasets 1-4. Each dot represents a variable in the data. Ideally, a perfect linear relationship between percentages in the real and synthesized data should be observed, such that points fall on the grey dashed line.

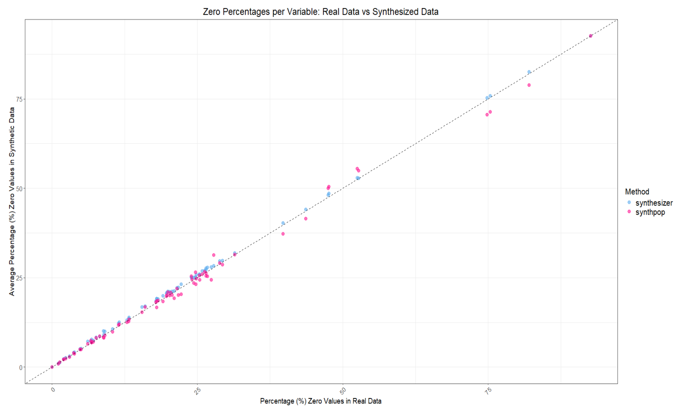


(a) Dataset 5

(b) Dataset 6

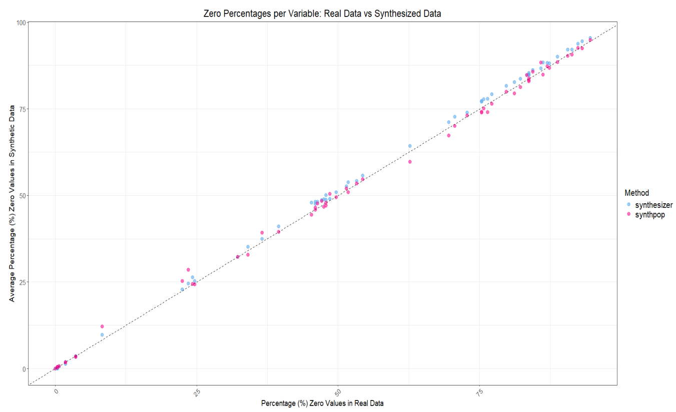


(c) Dataset 7



(d) Dataset 8

(e) Dataset 9



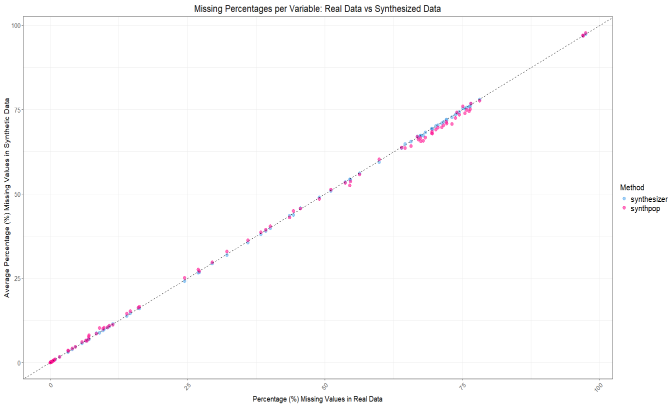
(f) Dataset 10

Figure 30: Zero Percentages in Real and Synthesized Datasets using both `synthesizer` and `synthpop` for real datasets 5-10. Each dot represents a variable in the data. Ideally, a perfect linear relationship between percentages in the real and synthesized data should be observed, such that points fall on the grey dashed line.

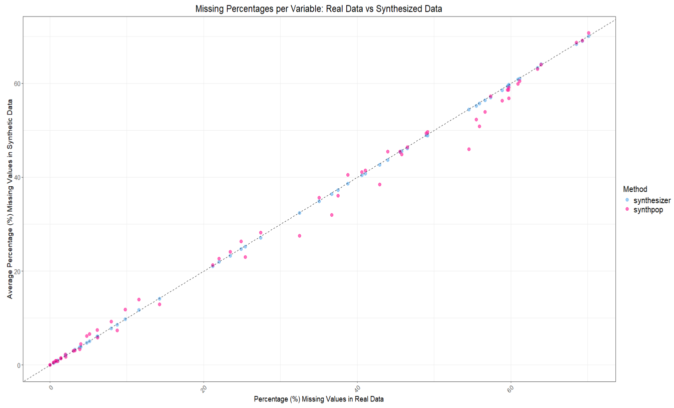
Appendix B

Percentage Missing Value Plots

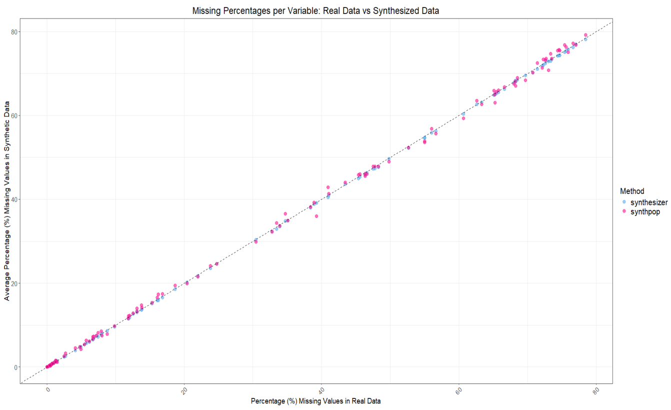
This appendix presents the plots of the results for the percentage of missing values in the synthesized datasets for all the real datasets.



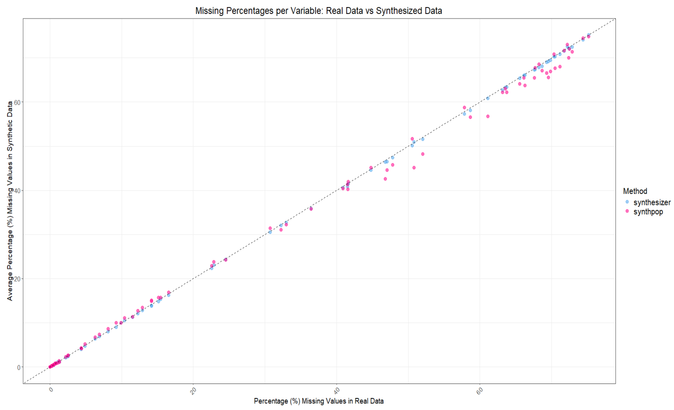
(a) Dataset 1



(b) Dataset 2

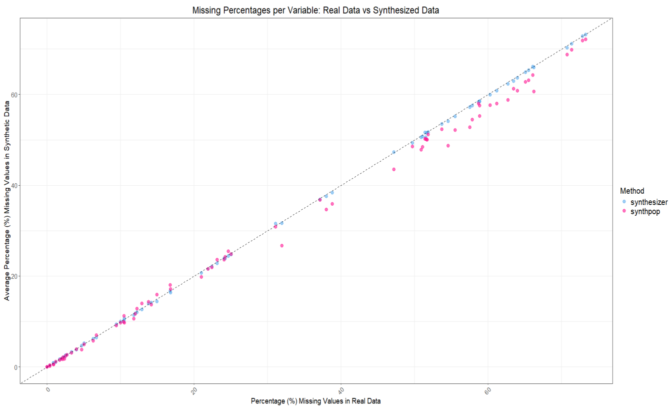


(c) Dataset 3

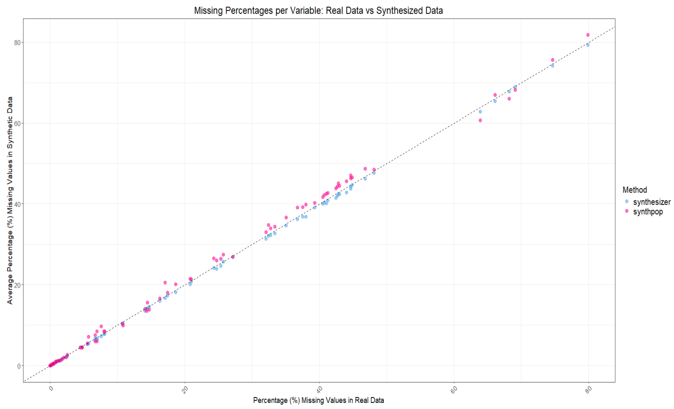


(d) Dataset 4

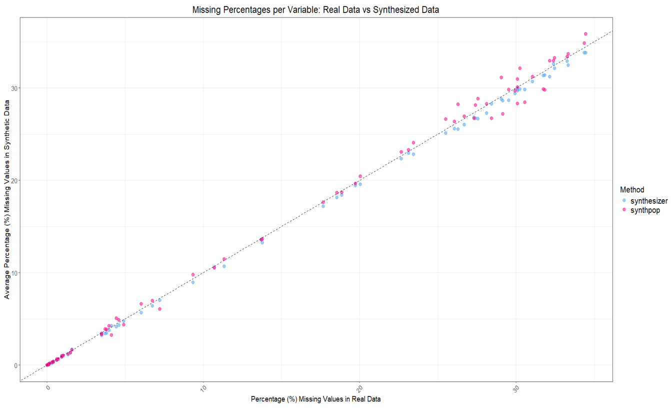
Figure 31: Missing Percentages in Real and Synthesized Datasets using both `synthesizer` and `synthpop` for real datasets 1-4. Each dot represents a variable in the data. Ideally, a perfect linear relationship between percentages in the real and synthesized data should be observed, such that points fall on the grey dashed line.



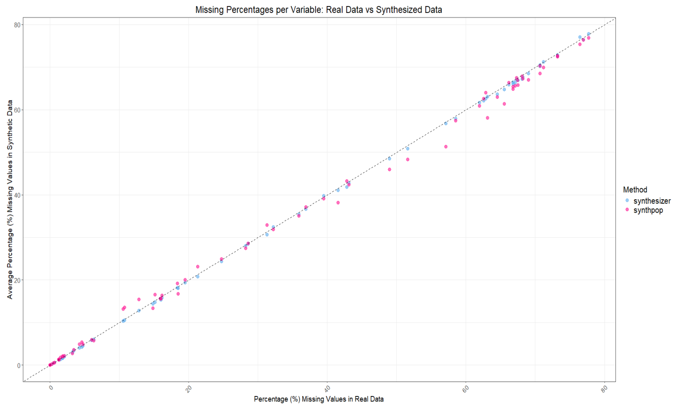
(a) Dataset 5



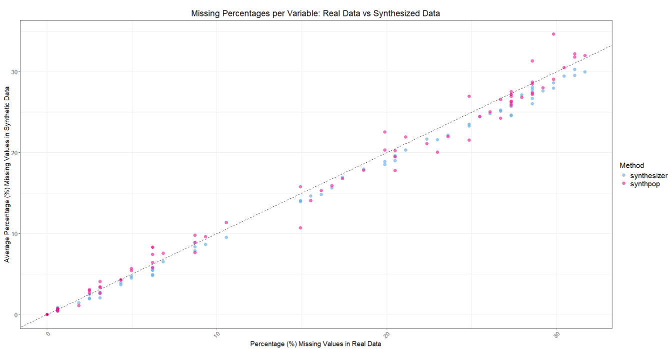
(b) Dataset 6



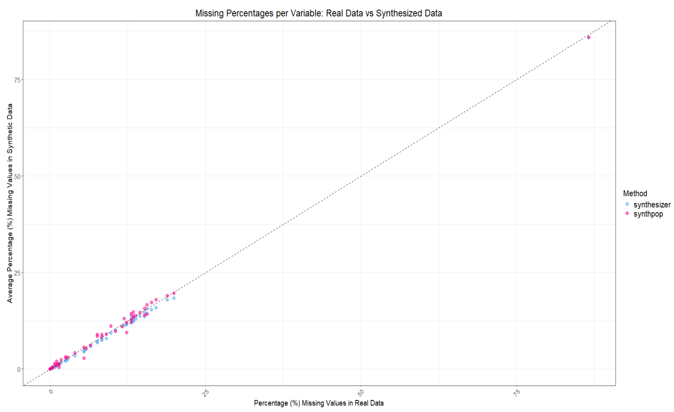
(c) Dataset 7



(d) Dataset 8



(e) Dataset 9



(f) Dataset 10

Figure 32: Missing Percentages in Real and Synthesized Datasets using both `synthesizer` and `synthpop` for real datasets 5-10. Each dot represents a variable in the data. Ideally, a perfect linear relationship between percentages in the real and synthesized data should be observed, such that points fall on the grey dashed line.

Appendix C

R Code

The R code for the computational study in this thesis is available online in a GitHub repository. See: `https://github.com/mishca-jacobs/MSc_Thesis_Code.git`