h_da
hochschule darmstadt
fachbereich mathematik
und naturwissenschaften

member of
EUROPEAN UNIVERSITY OF TECHNOLOGY

Masterarbeit im Studiengang Data Science

Potential of Synthetic Data for Providing Georeferenced Official Microdata

Abschlussarbeit zur Erlangung des akademischen Grades Master of Science (M. Sc.) im Studiengang Data Science

Hendrik Gressmann

22. März 2025

Das Thema stellte
Prof. Dr. Antje Jahn

Korreferent
Prof. Dr. Timo Schürg

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht. Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen. Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Darmstadt, 22. März 2025

Hendrik Gressmann

Abstract English

The Research Data Center of the Federal Statistical Office of Germany offers data access to German official microdata for independent research institutions. Depending on the selected way of data access, the strict data privacy regulations for German official microdata mostly require sensitive attributes, such as precise geospatial information, to be anonymized, which makes the data products unusable for precise geospatial analysis. However, many research institutes might heavily benefit from research data with precise geo-referencing. Targeting this conflict, in this thesis paper we present and empirically examine five different data synthesis methods on the German Census Data from 2011, with the goal of finding a way for creating and providing useful synthetic versions of German official microdata with precise geospatial attributes. Regarding the synthesis models, we apply two parametric synthesizers, which are a Copula Synthesizer with Frequency Encoding and a Copula Synthesizer with One-Hot Encoding, as well as two synthesizers without parametric assumptions, which are a Random Forest Synthesizer and a CART (Classification and Regression Tree) Synthesizer, and finally Geomasking, a commonly used anonymization method for georeferenced data. We assess the potential the different synthesizers provide us using various metrics from the field of data synthesis for quantifying data utility and data privacy. The result of this thesis is that out of all examined synthesizers, the CART Synthesizer performs the best in terms of data utility, while offering a negligible risk in regards to data privacy. It would be conceivable to make such a synthetic data set with precise geospatial information accessible for independent research institutions via the remote access system of the Research Data Center.

Abstract Deutsch

Das Forschungsdatenzentrum des Statistischen Bundesamtes kann unabhängigen Forschungseinrichtungen Zugang zu amtlichen Einzeldaten gewähren. Je nach gewähltem Datenzugangsweg verlangen die strengen Datenschutzbestimmungen für amtliche Datenprodukte in der Regel, dass präzise räumliche Attribute, wie andere sensible Attribute auch, anonymisiert werden, was die Daten für Zwecke räumlicher Analysen unbrauchbar macht. Einige Forschungseinrichtungen könnten jedoch stark von Forschungsdaten mit präziser Georeferenzierung profitieren. Um diesem Konflikt Abhilfe zu verschaffen, präsentieren und analysieren wir in dieser Thesis fünf verschiedene Methoden der Datensynthese anhand der deutschen Zensusdaten von 2011, mit dem Ziel der Erstellung und Bereitstellung brauchbarer synthetischer Versionen amtlicher deutscher Datenprodukte mit präzisen räumlichen Attributen. Die angewendeten Synthesemodelle sind ein Copula Synthesizer mit Frequency Encoding und ein Copula Synthesizer mit One-Hot Encoding, welche beide parametrische Modelle sind, sowie ein Random Forest Synthesizer und ein CART (Classification and Regression Tree) Synthesizer, welche beide Synthesizer ohne parametrische Annahmen sind, und schließlich Geomasking, eine häufig verwendete Anonymisierungsmethode für georeferenzierte Daten. Wir bewerten das Potenzial der verschiedenen Synthesizer anhand verschiedener Metriken aus dem Bereich der Datensynthese zur Quantifizierung der Data Utility und Data Privacy. Das Ergebnis dieser Thesis ist, dass von allen untersuchten Synthesizern der CART Synthesizer in Bezug auf Data Utility am besten performt und gleichzeitig ein vernachlässigbares Risiko bezüglich Data Privacy aufweist. Es wäre denkbar, einen solchen synthetischen Datensatz über das Remote Access System des Forschungsdatenzentrums für unabhängige Forschungseinrichtungen zugänglich zu machen.

Table of Contents

1 Introduction

Statistical microdata plays an essential role in knowledge generation about various phenomena in and among society. An example of such extremely large and thorough microdata is census data. The research of independent scientific institutions can profit heavily from census data products, as they include microdata from up to 100% of a country's population with a huge number of attributes. For some scientific questions geospatial analysis might be required, for which the geospatial attributes of the census products in combination with other non-spatial census attributes can be used. Moreover, the census data might be linked to auxiliary data, using a matching process relying on the congruence of their geographic links. This procedure makes geographically detailed analysis in other scientific fields possible. The authors of [T K23] demonstrate this by calculating a regionally disaggregated metric (Necesidades Básicas Insatisfechas (NBI)) that is officially used as national poverty index in Costa Rica. The more precisely the housing location of the survey participants is known, the more accurate the results of the respective research will be.

Since census data consists of the private microdata of the survey participants, sensitive attributes, like name and address, are encrypted or stripped off in the published data sets. This generally does not affect the utility of the data. Also, low-level geospatial information might be sensitive because especially in combination with other census attributes it could help a potential attacker to re-identify individuals among the data set. The more granular these locations are listed, the easier it is to re-identify individuals and therefore access their personal confidential data. In order to keep this re-identification risk low, spatial data should only be listed on a reasonably high level.

Considering both views on the release of geospatial census data, the utility is conflicting with the privacy demands of the individuals who have participated in the survey, as for instance profoundly described in [DH23].

In order to protect the confidentiality of sensitive microdata while still preserving its usefulness, scientists came up with a promising trade-off: Instead of releasing the original data and risking privacy issues, they release synthetic data. Data providers fit a model to the original data, sample values from this model and finally use them to replace the original values. The idea of synthetic data for the purpose of disclosure avoidance is according to [DH23] commonly attributed to [Rub93] and [Lit93]. Those first approaches were based on multiple imputation, similarly as for imputing missing values. [Rub93] suggested to treat all values of the data set as missing values, which were to be replaced by samples of the imputation model trained on the original data set. This method is considered to be a full synthesis because every attribute of the original data set is synthesized. The level of protection is very high because not only is none of the original values present in the synthetic data set anymore, also there is no one-to-one relation between one synthetic and one original record. The quality of such data, however, strongly depends on the quality of the imputation model. It can be challenging to

find a model that would be able to preserve the complex relations between different variables and simultaneously consider various and convoluted logical constraints. The approach by [Lit93] is closely related and creates so-called partial data synthesis, overcoming some of the challenges of fully synthetic data. The idea is to only treat some attribute values as missing, for instance, those at especially high risk, and thus only impute those attributes with samples from an imputation model. This approach allows a higher level of flexibility, since data providers can themselves decide which of the attributes to synthesize, based on their requirements.

The partial synthesis approach is nowadays commonly used to solve the privacy-utility-conflict around geospatial microdata by only treating the sensitive geospatial attributes, as explained in [DH20]. One group of such methods would include combinations of deletion and perturbation procedures of the sensitive attributes, or replacing the true location of an individual with aggregated (i.e. area-level), and possibly even randomized information. The advantages of this procedure are that the quality of the remaining (non-spatial) survey information is not affected. However, it often fails to provide a sufficient level of privacy protection, since already small subsets of original attribute values can increase the risk of re-identification, even in incomplete, pseudonymous datasets. Another way of synthesizing the geospatial attribute is by recycling the fundamental ideas of data synthesis and fitting machine learning models to the data. [DH20] compared a semi-parametric model based on a Bayesian procedure to two CART (Classification and Regression Tree) Models. The study results imply that the non-parametric CART Models generally performed better. This way of treating the sensitive geo-attribute is supposed to allow better preservation of the original distribution, as well as a better control over the re-identification risk. However, the geographic links to potential auxiliary data get destroyed.

Addressing especially this last mentioned issue, [T K23] recently proposed a fundamentally different microdata dissemination strategy. Their main idea was to publish two datasets, instead of just one. The first one D_{no} is the original data with the sensitive geospatial identifiers stripped off. The second one D_{syn} consists of the synthetic microdata, while the geospatial identifiers remain untouched. They justify their new approach with a more user-centric perspective on synthetic data. Some analysis on household surveys requires precise and representative data, while geospatial features are only to be considered on a high regional level, if at all. In this case, the data set D_{no} can be used. Other kinds of analysis requires low-level geospatial information, especially as a congruent link to auxiliary data, for which D_{syn} may be used. Their experiments of data synthesis and the calculation of quantifiable metrics for privacy and utility were conducted on the Costa Rica Census Data from 2011. The authors claim that their method reduces the re-identification risk and increases data utility for spatial analysis, compared to other currently used solutions.

In Germany, official data is created and maintained by the Federal Statistical Office of Germany (Statistisches Bundesamt). Based on §16 BStatG (Bundesstatistikgesetz), the Federal Statistical Office is legally required to provide their official data sources for research purposes to independent research institutions, which is offered through the Research Data Center of this institution. At the same time, the legislation requires the Federal Statistical Office to protect the personal data of survey participants from disclosure. Since the currently used anonymization methods for personal microdata only dissolve this conflict partially, the institution is sincerely interested in new advances in the field of data synthesis for the

anonymization of sensitive data sources.

In this project, we will apply, evaluate, and compare five different data synthesis methods. At first, we will examine the Copula Synthesizer with Frequency Encoding, as proposed by [T K23]. Here, we will especially focus on replicating their work as closely as possible and will compare our findings on this synthesis method with their findings. Secondly, we will perform another Copula Synthesis, but with One-Hot encoding. The third synthesizer used by us will be a CART Synthesizer, which is based on classification trees, while the fourth synthesizer will be a Random Forest Synthesizer, based on multiple Random Forests. As the final synthesis method, we will apply Geomasking, which was also carried out in [T K23]. In order to evaluate the potential of the different synthesizers for German official statistics and science, we will apply those five synthesizers to the German Census Data from 2011. Finally, the five different approaches will be discussed and evaluated, especially considering the high demands on data security for German official data.

2 Data Synthesis

2.1 Administrative Geo-Spatial Structure Germany and Geo-Spatial attributes in German Census Data

For understanding how the different synthesis algorithms can be applied to the German Census Data, we first have to define the geospatial structure of levels of administrative areas in Germany. For simplification reasons, we will only mention the administration levels relevant for this project.

Germany is split into 16 federal states (Bundesländer). One level below the federal states are the 412 counties and cities [1] (Kreise und kreisfreie Städte). On the lowest administration level worth mentioning here are the communes (Gemeinden und Gemeindeverbände), of which there are over 11,000 in the entire country.

For our project, we use 26 attributes of the German Census Data. The data set comprises 80,209,997 observations, one row or entry for every person living in Germany at the reference date. In the following, one entry will be referred to as a record. Besides other attributes, one important piece of information derivable from the Census Data is the home location of an individual, which in the data is defined by two groups of geospatial attributes. One of those groups consist of the geospatial administrative area types, which are the federal states, the counties or cities and the communes, as described above. Defining the commune implicitly also defines the county or city and the federal state of an individual. For the other group of geospatial attributes, Germany is split up into grid cells of 10x10, 1x1 and 0.1x0.1 kilometers. For all three of those levels, there is a separate attribute in the data revealing which grid cells an individual is located in. All of the mentioned spatial attributes will be relevant later on in section 5.2.2. However, for understanding how we apply the synthesis algorithms to the census data, only the counties and cities as well as the 10x10km grid cells are of relevance. The relationship between those two attributes is visually presented in figure 2.1. We can clearly see that there is no hierarchy defined between the 10x10km grid cells and the counties and cities, since one county or city can include multiple 10x10km grid cells and vice versa.

[1] More precisely: counties and cities not associated with a county

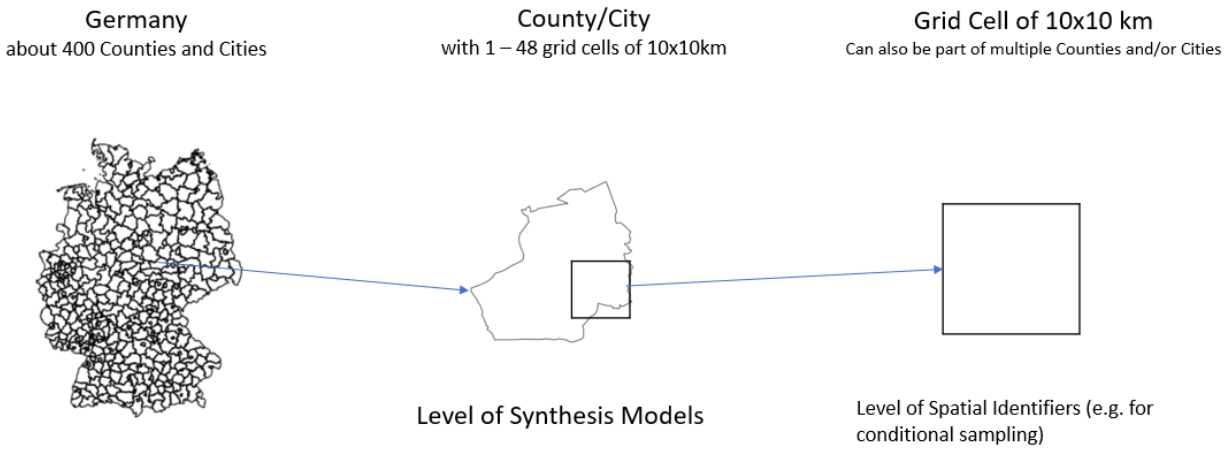


Figure 2.1: Administrative Disaggregation Structure Germany (simplified)

Furthermore, additional details on our data source will be discussed later on in chapter 5. A tabled summary including a semantic description of the different variables can be found in the appendix 7.1.

2.2 Algorithms for Data Synthesis

Before presenting the algorithms, we will introduce a few general definitions. If not stated otherwise, we will stick to those definitions throughout this entire document.

- The original data set D_{orig} is the Census Data after cleaning and feature selection, as described in chapter 5.
- The synthetic data set D_{syn} is the product of the data synthesis, based on D_{orig} and any synthesis algorithm.
- If D_{syn} is derived from a specific synthesis method, we will mark this with the respective subscript:
  - Copula Synthesis with Frequency Encoding: D_{copFE}
  - Copula Synthesis with One Hot Encoding: D_{copOH}
  - Random Forest Synthesizer: D_{rf}
  - CART Synthesizer: D_{cart}
  - Geomasking: D_{geo}
- Both Copula Synthesizers are based on a multivariate normal distribution. One aspect of normal distributions is that they are only defined for numerical values. However, our data source mainly comprises categorical attributes, which requires some data transformation via encoding. The encoded versions of any data set will be marked by a tilde: $\tilde{D}_{orig}, \tilde{D}_{copFE}, \tilde{D}_{copOH}$

- The counties and cities will be our strata s. For every stratum, we will apply the synthesis algorithm separately and therefore train 412 separate models for each synthesis method we want to evaluate.
- The identifier for the individuals will be i. The respective records will be D_i. Therefore $D_{orig,i}$ is record i of D_{orig}, in the same way $D_{orig,s}$ is all data from stratum s of D_{orig}.
- The identifier for the attributes will be noted as j. The respective attributes will be noted as D^j. Therefore, D^j_{orig} is attribute j of D_{orig} and $D^j_{orig,s}$ is attribute j of $D_{orig,s}$.
- We will use the 10x10km grid cells as spatial identifiers. The respective attribute in the data source is called GITTER_ID_10KM, but we will mostly be referring to it as *loc*. Since we will train a different synthesis model for every stratum s, the same spatial identifier may be part of multiple strata.
- The spatial identifier *loc* is the only attribute that does not get synthesized (exception: Geomasking). Therefore, $D_{orig,loc}$=$D_{syn,loc}$.
- As already mentioned in the introduction, the second data set of the final data product, which is the original data without spatial identifier D^{-j}_{orig}, will be noted as D_{no}.

2.2.1 Copula Synthesis with Frequency Encoding

The Copula Synthesis with Frequency Encoding is the algorithm suggested by [T K23]. The main idea of the Copula Synthesis is to sample synthetic records from a multivariate normal distribution that was estimated on D_{orig}. [T K23] name this synthesis model Copula Synthesis because they describe the sampling process from the synthesis model as sampling from a multivariate Gaussian copula, as defined by [Skl59]. Those samples are the quantiles that later create the synthetic samples by inserting them into the distribution function of a multivariate normal distribution.

One aspect of normal distributions to consider is that they are only defined for numerical values. However, our data source and also most census data in general is largely made up of categorical variables. Thus, some data transformation is required. [T K23] uses Frequency Encoding for the transformation from D_{orig} to $\tilde{D}_{orig}$, while mentioning that One-Hot Encoding could be applied instead. The method of Frequency Encoding used here is borrowed from [Man77] and is applied separately to every attribute with categorical values. For every attribute, also for $D_{orig,loc}$, the Frequency Encoding algorithm splits up the interval $[0, 1]$ into smaller intervals I_c and assigns each I_c to a class c. The sizes of I_c are proportional to the sizes of c, moreover, the intervals I_c get ordered by size. Consequently, the smallest intervals are close to zero, while the largest intervals are close to one. Then, it uses the mean value of each I_c as encoded value of the respective class c. The result of this encoding process is the encoded data set $\tilde{D}_{orig}$. Since the data synthesis is carried out for each stratum separately, we will also apply the encoding process for every stratum separately.

The basic idea behind the synthesis algorithm is that besides $\tilde{D}_{orig,loc}$, the other attributes are sampled from a multivariate normal distribution that is estimated from $\tilde{D}_{orig}$, with the sampling being conditioned on $\tilde{D}_{orig,loc}$. We need to consider that sampled values from a multivariate Gaussian normal distribution might not always follow logical characteristics

and relations that we expect from real word data. Thus, we will impose constraints on the synthetic values (e.g. that a person's age should not be negative), which are applied via rejection sampling. That means, if a sample w does not meet the constraints, new samples w are drawn until the constraints are met. Details on the imposed constraints can be found in the appendix 7.2. In algorithm 1, one will find the pseudo code of the copula algorithm.

Algorithm 1 Copula-based data synthesis

Input $\tilde{D}_{orig}$
Output $\tilde{D}_{copFE}$

for $s \in \tilde{D}_{orig}$ **do**
  Λ : Joint distribution of $\tilde{D}_{orig}$ with $\Lambda \sim \mathcal{N}(\mu_\Lambda, \Sigma_\Lambda)$
  $F_\Lambda(x)$: CDF (Cumulative Density Function) of Λ
  for $i \in s$ **do**
    while w does not meet constraints **do**
      $w \sim F_\Lambda\left(x | x_0 = \tilde{D}_{origFE,loc}\right)$ ▷ Sampling conditioned on spatial identifier
    end while
    $\tilde{D}_{syn,i} \leftarrow w$ ▷ The Sample is the new synthetic record
  end for
end for

For the re-transformation from $\tilde{D}_{copFE}$ to D_{copFE}, the algorithm assigns a new categorical value for each originally categorical attribute. Using the intervals I_c as obtained in the encoding process, the decoding process checks for every record i from $\tilde{D}_{copFE}$ into which of those intervals the numeric values fall. This decides, which class c to assign as new synthetic value. The final result is the synthetic data set D_{copFE}.

2.2.2 Copula Synthesis with One-Hot Encoding

In this section, we will examine some of the features of the Copula Synthesis with Frequency Encoding, and point out why this encoding method might suffer from some weaknesses. We will furthermore introduce and discuss One-Hot Encoding as a potential alternative encoding, as well as necessary alternations to the synthesis algorithm due to the use of One-Hot Encoding instead of Frequency Encoding.

Potential Weaknesses of the Copula Synthesis with Frequency Encoding

For analyzing how the Copula Synthesis models frequency encoded attributes, we will take a look at a density histogram of three frequency encoded variables chosen for demonstration purposes, FAMSTAND_AUSF, MHGLAND_KONT_MR and HH_STATUS_NAT (details on the variables in table 7.1), together with their respective estimated Gaussian curves, which is how the Copula Synthesizer models the distributions. To allow a better visual comparison, the histogram bars are scaled to the height of the highest bar being $h_{max} = 1$.

When taking a closer look at the algorithm of the Frequency Encoding, we will find that before the encoding takes place, the interval between 0 and 1 is divided into intervals I_i of length l_i, proportional to the size of class i. Therefore, after synthesis, a hypothetic random

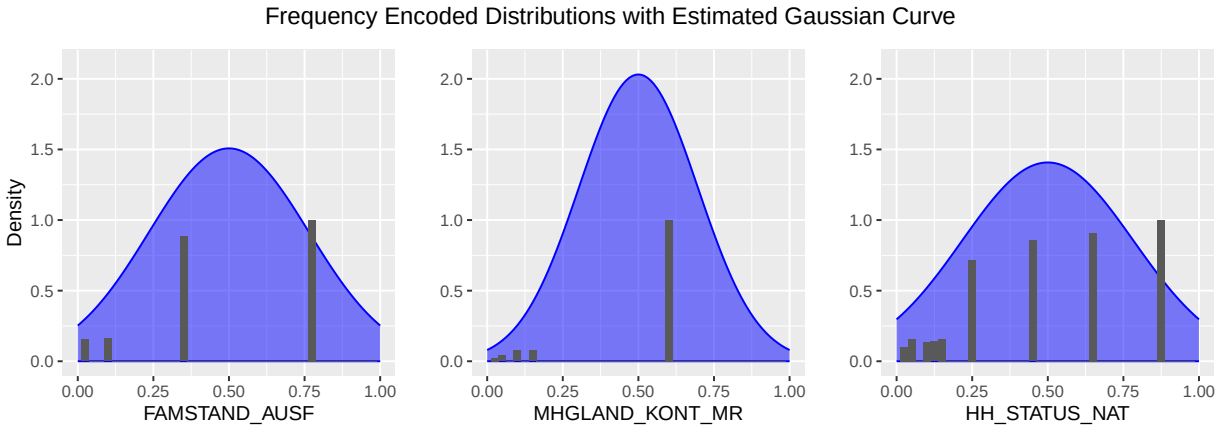


Figure 2.2: Distribution of Frequency Encoded Variables with their Gaussian Curves

sample of $X \sim U[0,1]$ has the probability of $p = l_i$ ending up in the interval I_i and being decoded into class i. This is exactly how we would want the records from $\tilde{D}_{copFE}$ to behave because the original distribution would be preserved. We would therefore have to assume a multivariate uniform distribution for modeling $\tilde{D}_{orig}$. This means that for all categorical and therefore frequency encoded variables, there is a discrepancy between the normal distribution assumed by the synthesis model and the actual, uniform distribution of the frequency encoded data.

Although the frequency encoded categorical variables follow a uniform distribution, it might not be sufficient to simply replace the multivariate normal distribution from the synthesis model with a multivariate uniform distribution. To understand why, we will take a look at figure 2.3 with the joint distribution of FAMSTAND_AUSF and HH_STATUS_NAT as an example, together with the resulting bivariate Gaussian density function estimated by the synthesis model.

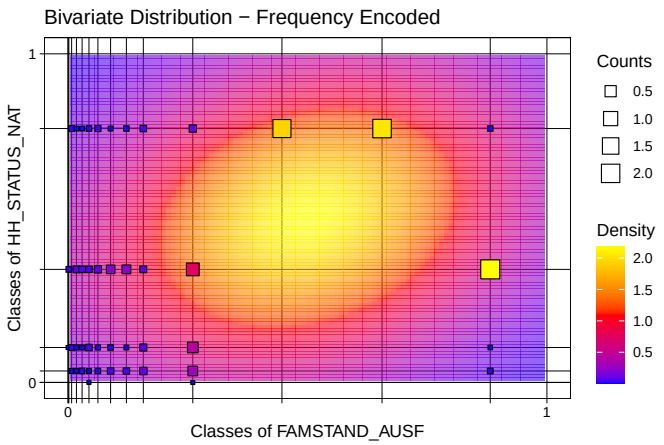


Figure 2.3: Bivariate Distribution of Frequency Encoded Attributes with Gaussian Curve

On the two axes we can find the classes of both variables after Frequency Encoding. We can see how they divide the interval between 0 and 1 into intervals proportional to their class size. The frequency encoded values are presented as a 2-dimensional histogram. Color and size of the squares represent the mass and are normalized to their highest value being equal to the maximum value of the Gaussian density function. As expected due to the ordering of the classes, the highest absolute frequencies are to be found in the upper right corner. It is clearly visible that the highest

values of the Gaussian density function are in the center of the figure, at the point (0.5|0.5). However, the empirical distribution of the bivariate frequency encoded variable has mass zero at values closest to (0.5|0.5). Generally, the second and third most frequent classes of FAMSTAND_AUSF only have a mass greater than zero for the most frequent class of HH_STATUS_NAT. Consequently, the empirical distribution of the samples from the underlying Gaussian density function does not reflect the empirical distribution of the original data well. As already mentioned, the multivariate Gaussian distribution does not have very flexible modeling properties. There is only one parameter that considers cross-variable relations, which is the covariance. Geometrically speaking, for a bivariate distribution it can only model the density distributions as an ellipsis with the center at (0.5|0.5). A multivariate uniform distribution has no parameters to describe relations between different variables at all, which makes it even less flexible than the normal distribution. And unless the classes of both variables occur independently from each other, which is usually not the case, the true relations of the different classes cannot be modeled.

To sum up, we are looking for an encoding method that fits to the assumed distribution, while also taking into account the individual relations of the different classes across categorical attributes.

One-Hot as potentially more suitable alternative to Frequency Encoding

One-Hot Encoding was also considered by [T K23], but eventually dismissed because of the high computation costs resulting from the huge amount of additional variables. These costs would still be high for our data source with 26 attributes, but more on a manageable level than for the data source from [T K23].

We will choose this encoding strategy as alternative to Frequency Encoding because it matches both requirements defined in section 2.2.2. Firstly, we can individually estimate a correlation coefficient for every combination of classes between different variables. If for instance a certain combination of two classes is rarely or never present in D_{orig}, the resulting correlation coefficient will be negative, which will make this combination occur rarely in D_{syn}. Secondly, One-Hot Encoding only creates the values 0 and 1. And even though a normal distribution assumption would not be justified, it might still be more adequate than for the frequency encoded variables.

Algorithm Copula Synthesis with One-Hot Encoding

In the following section, we will focus on the aspects of the Copula Synthesis that are different from the Copula Synthesis with Frequency Encoding. Apart from those differences, the Copula Synthesis with One-Hot Encoding is applied the same way as in section 2.2.1. Since the entire data source has 3826 classes in the spatial attribute GITTER___ID_10KM, but only up to 48 classes per stratum, we will apply the One-Hot Encoding for every stratum separately and therefore keep the number of new attributes created by the One-Hot Encoding per data set at 48 or below.

One problem that One-Hot Encoding causes are potential multicollinearities. Consequently, the estimated covariance matrix is not positive definite, which disables the creation of a

numerically stable distribution function for drawing the synthetic samples. In order to avoid this issue, the following procedures will be applied:

We encode D_{orig} by only creating $n^j_{classes} - 1$ encoded variables for each variable j, thus, the resulting matrix $\tilde{D}_{orig}$ has full rank. By encoding every stratum separately, we avoid that certain strata of $\tilde{D}_{orig}$ have attributes that only contain the value zero and thereby avoid the creation of singular columns. Another issue are the missing values for the three categorical attributes RAUMANZAHL_KLASS, HH_GROESSE_PERSON and GEBTYPGROESSE, as they all occur for the exact same records (more details on the missing values in section 5.1.3). Since missing values in categorical attributes are considered a separate class, $\tilde{D}_{orig}$ will include three identical columns. This issue is treated by using these missing values as the class that does not receive its own variable and is therefore only represented as a row of zeros. Despite applying those methods, the resulting covariance matrix is not positive definite. A specific reason is not determinable, it therefore must be caused by multicollinearities in higher dimensions.

Figure 2.4 illustrates how many strata have a matrix $\tilde{D}_{orig}$ that is affected by multicollinearities. The shared horizontal axis has no unit, while the vertical axis represents the number of strata. The histogram on the top with the title *Eigenvalues* counts the number of eigenvalues lower than 10^{-8} for every stratum, under application of the procedures explained above. The histogram with the title *Identical* shows the number of attributes with correlation coefficient $\rho = 1$ and therefore effectively the number of identical attributes. One could argue that such duplicate columns could simply be removed from the data source and added again after synthesis, using the synthetic values of the identical attribute that was not deleted. However, in the last histogram with the title *Differences*, where we present the stratum-wise difference of the first two histograms (*Eigenvalues*−*Identical*), we can see that there are a lot of strata where the number of eigenvalues lower than 10^{-8} is higher than the number of correlation coefficients $\rho = 1$. The respective area is marked with a red circle. Removing duplicate columns would therefore in many cases not solve the multicollinearity issue. The differences < 0 can occur if more than two attributes are equal. Three identical columns for instance produce three covariances with $\rho = 1$, while only causing two eigenvalues around zero.

We will now use a different approach, based on the following idea: Instead of trying to obtain a positive definite covariance matrix, we will tweak the covariance matrix in order to make it positive definite, even though the underlying data does not have full rank.

We start by calculating the normal distribution function $\overline{\Lambda}$, which uses a null vector $\vec{0}$ as mean values, the correlations of correlation matrix P_Λ as covariances, and $1 + \epsilon$ as variances. By adding ϵ to the diagonal of the covariance matrix $\Sigma_{\overline{\Lambda}}$, we make it positive definite. The higher the value ϵ, the better the numerical stability of $\overline{\Lambda}$ gets, but also the higher the bias of the variances gets. For the calculations in this project, we have to choose a value for ϵ and choose $\epsilon = 0.05$, as this value seems to be a good compromise between avoiding numerical instabilities and keeping the bias of the variance low. After sampling from $\overline{\Lambda}$, those samples $\overline{w}$ are being z-transformed, so that the marginal distributions (therefore μ_Λ and σ_Λ) agree with Λ. Now we check the same constraints for integer variables as described in section 2.2.1. If all constraints are met, w are the values of the new synthetic record. The pseudo code of the algorithm can be found in algorithm 2.

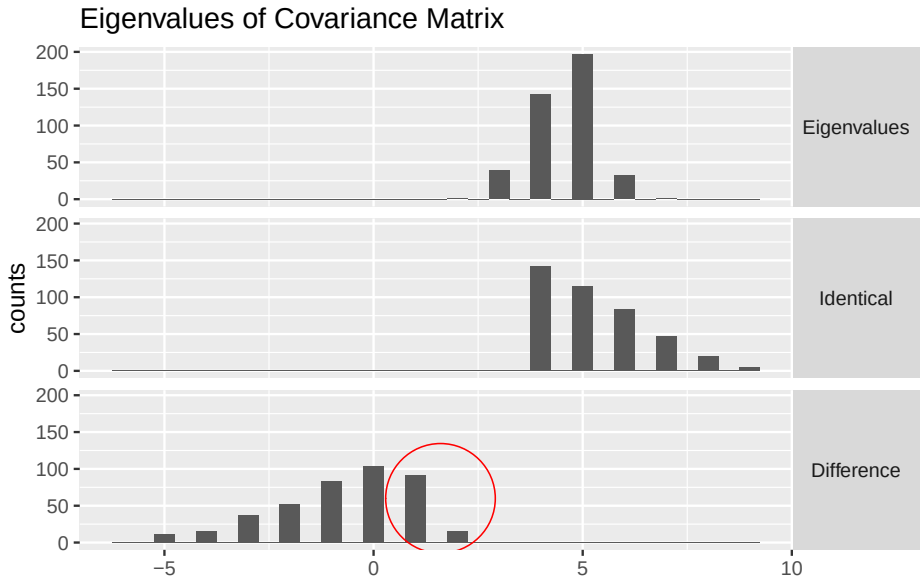


Figure 2.4: Eigenvalues equal to Zero, compared to Number of Identical Attributes

Algorithm 2 Copula-based data synthesis with One-Hot Encoded Data

Input $\tilde{D}_{orig}$
Output $\tilde{D}_{copOH}$

for $s \in \tilde{D}_{orig}$ **do**
  Λ : Joint distribution of $\tilde{D}_{orig,s}$ with $\Lambda \sim \mathcal{N}(\mu_\Lambda, \Sigma_\Lambda)$
  μ_Λ : Marginal Mean Values of $\tilde{D}_{orig,s}$
  σ_Λ : Marginal Variances of $\tilde{D}_{orig,s}$
  P_Λ : Correlation Matrix of $\tilde{D}_{orig,s}$
  $\overline{P}_\Lambda \leftarrow P_\Lambda + I \cdot (1+\epsilon)$ ▷ Add ϵ to avoid numerical instabilities
  $\overline{\Lambda}_0 \sim \mathcal{N}\left(\vec{0}, \overline{P}_\Lambda\right)$, $F_{\overline{\Lambda}_0}(x)$ as CDF *(Cumulative Density Function)* of $\overline{\Lambda}_0$
  for $i \in s$ **do**
    while w does not meet constraints **do**
      $\overline{w}_0 \sim F_{\overline{\Lambda}_0}\left(x | x_{1:n_{class}} = \tilde{D}_{orig,loc}\right)$ ▷ $\tilde{D}_{orig,loc}$ is a matrix with n_{class} columns, which is the number of classes in $D_{orig,loc}$
      $w \leftarrow \overline{w}_0 \cdot \frac{\sigma_\Lambda}{\sqrt{1+\epsilon}} + \mu_\Lambda$ ▷ Z-Transformation of marginal distributions
    end while
    $\tilde{D}_{copOH,i} \leftarrow w$ ▷ The Sample is the new synthetic record
  end for
end for

Since we artificially increase the main diagonal of the covariance matrix (multiplication with $1 + \epsilon$) before sampling, while at the same time leaving the covariances unchanged and later transform the samples (division with $1 + \epsilon$), the resulting covariances in D_{syn} are expected to be smaller than in D_{orig}, by $\rho_{orig} \cdot \frac{1}{1+\epsilon} = \rho_{syn}$. An alternative approach could have been to preserve the covariances by simply not dividing $\overline{w}_0$ by $1 + \epsilon$. However, we considered it more important for the variances to be accurately preserved, since this at least leads to an exact preservation of the marginal distributions, which would be useful for potential analysis of the synthetic data with focus on the univariate distributions.

Decoder for reversing the One-Hot Encoding

Despite One-Hot Encoding being one of the most common methods to enable the use of categorical attributes in a numerical model, applications where the transformation is reversed by decoding appear to be relatively rare. The general issue is that the resulting samples forming the data set $\tilde{D}_{copOH}$ are not binary, but numeric. Therefore, we need to find a decoder that chooses the appropriate attribute for every encoded variable and record, which then becomes the respective synthetic value. There are two different encoding schemes that we have considered:

- *Maximum*: For record i, choose the attribute where $\tilde{D}_{copOH,i}$ has the highest value and use the class represented by this attribute as synthetic values $D_{copOH,i}$.
- *Distribution*: Using $\tilde{D}_{copOH,i}$ as weights for a multinomial distribution from which $D_{copOH,i}$ is sampled. Classes with negative values are excluded since they would have negative probabilities.

We investigated both methods using two times 99 small artificial data sets, consisting of one categorical attribute with 2 and 11 classes, respectively. The relative size of our class of interest, α, among the 99 data sets was $[0.01, 0.02, ..., 0.99]$, while the other 1 or 10 classes were filling up to 100% (while being of equal size). We applied One-Hot Encoding without full rank to each data set, estimated a multivariate normal distribution, sampled $n = 10000$ records from it and finally applied both previously defined decoding mechanisms. The results are displayed in figure 2.5.

In the two different rows of the figure we can find the data sets with 11 classes in the upper row and with 2 classes in the bottom row, while the two different columns represent the two different decoders. The gray dashed lines represent the ratio of α before encoding and after decoding. A perfect encoder-decoder mechanism would produce synthetic data sets for which those two relative sizes are always equal, which is presented by the blue angle bisector. This means that we would like the line of the actual decoder to be as close as possible to the line of the perfect decoder.

Based on the first impression of figure 2.5, we would select decoder *Maximum* for our purpose. However, it is more interesting to focus on the performance of the low values (marked with a red circle). Most classes in our data source have relative sizes below 10%, which is also due to the fact that most attributes have a high number of classes. The problem with encoder *Maximum* is that the preserved likelihoods of those small classes are extremely low and for a relative size of 0.01 even completely vanish from the synthetic data. This appears to be especially critical for variables with a small number of classes, as to be found in the left

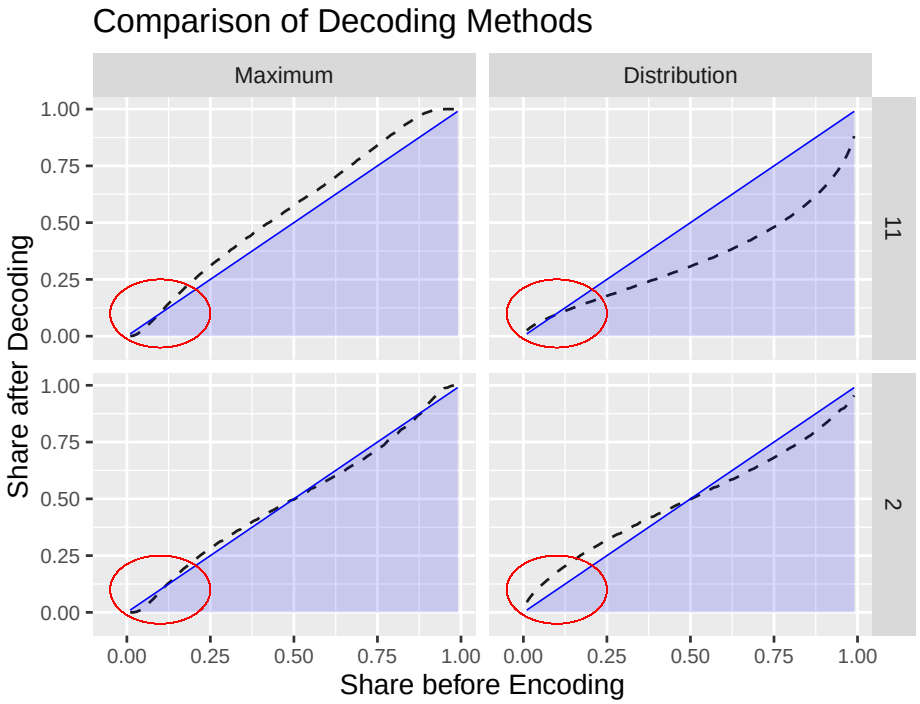


Figure 2.5: Potential of two different Decoders for Reversing One-Hot Encoding for our Purpose. Upper Row: Plots for 11 Classes, Bottom Row: Plots for 2 Classes

bottom plot of the figure. The results of *Distribution* might overall not look as promising, but this decoder seems to preserve the relative sizes of those small classes still much better, although generating them a bit too big. To conclude, we choose *Distribution* as our decoding mechanism for reversing the One-Hot Encoding.

2.2.3 Data Synthesis based on CART Models

In this section, we discuss the data synthesis based on CART (Classification and Regression Trees) after [Rei05]. Initially, we will briefly discuss the general weaknesses of parametric models on complex and mixed data sets and afterwards discuss how non-parametric synthesis model might be more suitable for our purposes.

Potential Weaknesses of Non-Parametric Models for Data Synthesis in general

In order to point out potential weaknesses of non-parametric models for data synthesis, we will take a look at an example of a density histogram of two integer variables, GEBTYPGROESSE and HH_GROESSE_PERSONEN, together with their estimated Gaussian curves. Figure 2.6 illustrates, how the actual distribution of GEBTYPGROESSE does not fit the respective Gaussian curve well, unlike the one of HH_GROESSE_PERSONEN. The restrictions in modeling ability of the Copula Synthesis due to the parametric assumptions could potentially be a severe problem, depending on the actual distributions of the data. This problem might especially be severe for numeric attributes, since those might follow any marginal distributions possible.

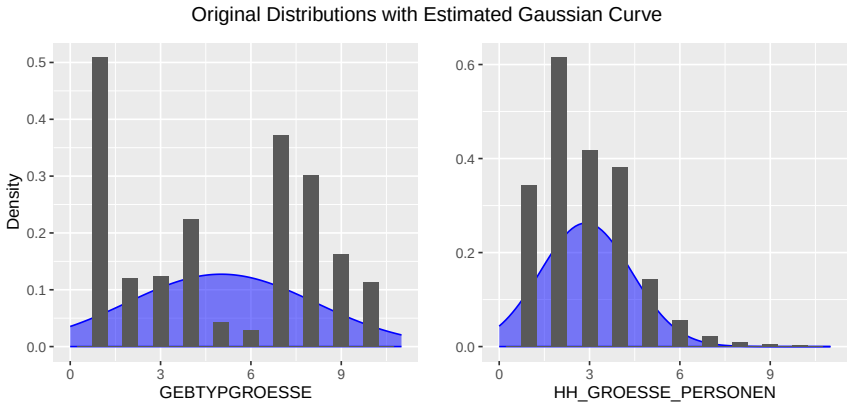


Figure 2.6: Distributions of Integer Variables with their Gaussian Curves

CART Synthesizer as promising alternative to Copula Synthesizers

As an alternative to the Copula Synthesizers, CART Synthesizers by [Rei05] do not come with any of the previously mentioned issues and are moreover highly popular in the field of synthetic data [DH23]. The advantages of these models for data synthesis are the following: Besides not requiring parametric assumptions they can model complex non-linear distributions and perform semi-automatic variable importance selection, by being trained to only split on relevant variables. CART, moreover, can be trained on a significantly smaller amount of data observations than other Machine Learning models without parametric assumptions, like Neural Networks, and generally require low computational costs. According to [DH23] and [DR11], CART Models are often said to outperform any other type of synthesis model. Another advantage is that we do not have to impose any constraints on the synthetic values because the synthesizer only produces values existing in the original data set. For instance, as long as in the original data the lowest number of rooms of any apartment is $n = 1$, there will be no synthetic record with a number of rooms of $n = 0$ or $n = -1$. Due to the advantages the CART based Synthesis provides, we will use it to investigate whether the utility of our synthetic data can be improved further, compared to the Copula Models.

Algorithm CART Synthesis

To adapt the algorithm to our requirements, we will re-use the idea from [T K23] to create two data sets, as also described in the introduction 1 : D_{syn}, where we synthesize all attributes, except for the spatial identifiers, and D_{no}, where we simply remove the spatial identifiers, but leave the other attributes unchanged.

Just like with the other synthesizers, we want to synthesize all attributes, except for *loc*. For synthesizing attribute j, a Classification or Regression Tree $\mathcal{T}_j$ is trained with the predictors being all remaining attributes that have either been synthesized already or are not supposed to be synthesized at all, while j is used as target variable. Then, for obtaining the synthetic value for record i, which therefore would be $D^j_{syn,i}$, we sample one record from all records in the same final node as i, using Bayesian Bootstrap [Rub81]. Attribute j of the sampled record is the new synthetic value. In case of a Regression Tree, we would have samples from an estimated kernel density distribution instead. Following this procedure, the model synthesizes

all attributes one by one. The pseudo code of this algorithm can be found in algorithm 3.

Algorithm 3 CART Data Synthesis

Input D_{orig}
Output D_{cart}

for $s \in D_{orig}$ **do**
  for $j \in D_{orig,s}$ **do**
    Train Classification Tree $\mathcal{T}_j$
    $D^j_{orig,s} \sim \mathcal{T}_j \left(D^{-j}_{orig,s} \right)$

    for $i \in s$ **do**
      $node_i$: Final Node of i
      $w \sim D^j_{orig,s} \in node_i$ ▷ Sampling using Bayesian Bootstrap
      $D^j_{cart,i} \leftarrow w$

    end for
  end for
end for

Regarding the synthesis order of the different attributes, [Rei05] presents two different approaches. The first option would be to use feature importance. We train a classification tree $\mathcal{T}_j$ for every candidate attribute j, with j_a as target attribute and all other attributes as predictors. The earlier a tree splits by a certain attribute, the higher its feature importance. The second option would be to synthesize the attributes by ascending order of the number of classes. Since this second option can speed up computation significantly, we choose to use this way of ordering for our project.

2.2.4 Data Synthesis based on Random Forests with OOB *(Out Of Bag)* Prediction

The standards for data privacy of German official micro data are very high and passing legal requirements for obtaining permission to release synthetic data with precise geospatial attributes is expected to be challenging. CART Synthesizers are generally found to perform best regarding the privacy-utility trade-off. However, [DR11] also finds that regarding data privacy, Random Forest Synthesizer ([CR10]) tend to outperform CART models. Since a special focus on data privacy is highly required for German official micro data, in this project we will also examine Random Forests Synthesizers. They generally provide similar advantages and disadvantages to CART Synthesizers. Despite being more expensive to run, which also depends on the hyperparameters and the size of the data source, Random Forests use a set of trees instead of just one, which might give them the ability to model even more complex attribute relationships than the CART Models.

Algorithm Random Forest Synthesis

The main idea is to train a Random Forest on the original data and then use predictions based on OOB votes for sampling the values of the new synthetic attribute.

Algorithm 4 Random Forest data synthesis

Input D_{orig}
Output D_{cart}

$(j_1, j_2..., j_{max})$: synthesis order of the different attributes

for $s \in D_{orig}$ **do**
  for $j_j \in (j_1, j_2..., j_{max})$ **do**
    Train Random Forest $\mathcal{T}_j$
    $D^j_{orig,s} \sim \mathcal{T}_j \left(D^{(j_{loc}, j_1, ..., j_{j-1})}_{orig,s} \right)$ with $\mathcal{T}_j = \left(\mathcal{T}_j^1, ..., \mathcal{T}_j^{t_{max}} \right)$, t_{max} the number of trees

    for $i \in s$ **do**
      Generate a vote v^t from every tree $\mathcal{T}_j^t$ in $\mathcal{T}_j$
      $(v^1, ..., v^{t_{max}}) \leftarrow \mathcal{T}_j \left(D^{(j_{loc}, j_1, ..., j_{j-1})}_{orig,s} \right)$

      Use the votes as probabilities for a multinomial distribution
      $M\left(v^1, ..., v^{t_{max}}\right) \leftarrow (v_1, ..., v^{t_{max}})$

      Use random sample from M as new synthetic value
      $w \sim M\left(v^1, ..., v^{t_{max}}\right)$
      $D^j_{rf,i} \leftarrow w$

    end for
  end for
end for

Just as CART Models, Random Forests can only have one target variable at a time, which means that we will again synthesize the attributes after each other. We start by synthesizing the first attribute j_1, by training a Random Forest $\mathcal{T}_1$ with loc as predictor and j_1 as target. Then we use every tree in $\mathcal{T}_1$ to predict votes $(v^1, ..., v^{t_{max}})$ based on loc for every record i individually. Based on the counts of those votes, we create a multinomial distribution and then randomly sample the synthetic values from it, which gives us D^1_{rf}. Now this process is repeated for j_2: We train a random forest $\mathcal{T}_2$ with j_1 and loc as predictors and j_2 as target. For predicting the values for the multinomial distribution, we use the already synthesized attributes. The idea behind this is to generate new synthetic observations with attribute values fitting to each other. This procedure is now also applied for $j_3, ..., j_{max}$ and we finally obtain the synthetic data set D_{rf}. The pseudo code of this algorithm can be found in algorithm 4.

One significant issue of Random Forest Synthesizers is that they tend to predict the synthetic values very close to the original values. This happens because the trees that have been trained on record i will usually predict the original value as new value for i. The resulting multinomial distributions would consequently be highly peaked around that original value. Without any further adjustment, the actual purpose of the data synthesis, which is masking D_{orig} to make it impossible to learn anything about certain individuals, would be missed. The authors of [CR10] suggest two ways of solving this issue.

The first one would be to include prior distributions to the multinomial distributions. This could ensure that there are only non-zero probabilities of generating any supported outcome values and would lower the peak around the class of the original value. The other one is to base the data for the multinomial distribution only on trees for which i does not appear in the training sample. Those trees are the so-called OOB trees. Their usual purpose is to allow calculating the OOB error for testing the performance of a Random Forest. Here, the OOB trees are simply used for prediction. We have to keep in mind that for data synthesis we cannot investigate the predicting abilities of our models because there is no quantifiable optimization criterion. The disadvantage of this method is that there is always a certain number of trees that will not be used for generating the multinomial distribution, which is why a high number of trees might be required. For solving this issue, we eventually chose an approach outside the proposals of [CR10]: We used approximate Random Forests by only using a very small sample size to train the trees. For approximate Random Forests, the R-function RandomForestSRC::rfscr.fast() for instance uses $n^{0.75}$ as default sample size. In the course of this, we increase the number of OOB trees to be close to the total number of trees, while at the same time limiting already high calculation costs significantly.

Regarding the synthesis order of the variables, the authors make the following suggestion, which we apply for this project: Different orderings might create different utility and risk profiles. However, if computation costs are a concern, one should order the variables by increasing number of categories. This means that the attributes with many categories are only included in the last Random Forests of a sequence, which can speed up the computation.

2.2.5 Geomasking

As mentioned in the introduction 1, Geomasking is a commonly used method for protecting micro data with sensitive spatial attributes. The authors of [T K23] used it as baseline model. The basic idea is that the locations get randomly altered within a certain radius, in order to obfuscate the true survey locations.

The Geomasking is the only algorithm in this project that is not applied stratum-wise. The prior location of i, r_i, is the geometric center of the attribute value of GITTER_ID_100x100m, in which the individual is located. First, two random samples from different uniform distributions are selected, creating angle $\phi \sim \mathcal{U}[0, 2\pi]$ and distance $d \sim \mathcal{U}[0, U_{upper}]$. The upper bound U_{upper} for the distance varies according to whether the record is located in a rural or an urban area. Afterwards, the new location r_i^{masked} is obtained by a shift of r_i, using the previously sampled values, as presented in equation 2.2.1.

$$r_{x,i}^{masked} \leftarrow r_{x,i} + s \cdot \cos(\phi) \tag{2.2.1}$$

$$r_{y,i}^{masked} \leftarrow r_{y,i} + s \cdot \sin(\phi) \tag{2.2.2}$$

Finally, the algorithm checks a constraint for the new location; it must still be within the borders of its original stratum s. If this is not the case, the process for record i is repeated. Using this constraint we make sure that the population structure within each county or city stays unmodified. After that process, the attribute *loc* of all records get assigned with their new spatial identifier. The new synthetic data set is called D_{geo}. Except for the spatial identifier, all other attribute values remain unchanged.

To find appropriate upper boundaries U_{upper} for sampling the distances d, in this project (and unlike in [T K23]) we decide to use a flexible upper boundary, depending on the population density of the respective stratum. In that way, we only need to deal with one hyper parameter, which is the number of other individuals N_{closer} who are expected to be closer to an individual's true location than its location in the synthetic data set. The dependency of N_{closer} and U_{upper} is presented in equation 2.2.3, with N_s being the number of inhabitants, A_s the area (in m^2) and D_s the population density (in $\frac{1}{m^2}$) of stratum s and $A_{Expected}$ being the expected area that is closer to an individual's true location than the individual itself is after Geomasking.

$$N_{closer} = A_{Expected} \cdot D_s = \pi(\frac{U_{upper}}{2})^2 \cdot \frac{N_s}{A_s} \tag{2.2.3}$$

$$U_{upper} = 2 * \sqrt{\frac{A_s}{N_s} \cdot \frac{N_{closer}}{\pi}} \tag{2.2.4}$$

3 Risk

The objective of the risk evaluation is to quantify the risk of a potential attacker being able to learn some information about individuals in the original data, based on the synthetic data and other prior knowledge. This is an unspecific definition. In reality, we can only quantify re-identification risks for a certain attacking scenario. The attacking scenarios may differ in the following aspects:

- Known key variables (attribute values the attacker already knows)
- Prior knowledge about the synthesis method
- Target variables (attributes the attacker is interested in)
- Whether we consider attribute disclosure (revealing information about attribute values) or identity disclosure (revealing information on whether a certain individual in present in the data set)
- Whether there is other information available that might help him find some attribute values? (In our case this would be D_{no})

In the following, we will present two different methods that help to get a good idea of the general re-identification risk of a synthetic data set. For all risk metrics calculated in this project, we assume that the attacker has both data sets (D_{syn} and D_{no}) available, since both of them are meant to be released at some point, even though they are not meant to be accessed simultaneously for research purposes.

3.1 Privacy Attack

The main idea of the Privacy Attack [T K23] is to train a model that predicts the sensitive attributes, based on some information that a potential attacker might have.

The authors of [T K23] suggest training a model f based on D_{syn}, with the sensitive attribute *loc* as target variable and the others as predictors. Then, based on D_{no}, they would try to predict the original spatial identifier D_{true}^{loc} using f and compare it to the true original spatial identifier. For f, the authors used a Random Forest Model with $n_{tree} = 500$ trees. Details on the other hyper-parameters are not given. Similarly, we use the R implementation RandomForestSRC::rfsrc() [Ish+08] with its default parameters, however with only 50 trees due to computational reasons.

For evaluating the results of the Privacy Attack we use the accuracy and the balanced accuracy. We define the accuracy *acc* in equation 3.1.1 and the balanced accuracy *bacc* in equation 3.1.2. Those metrics give us information on the risk of attribute disclosure regarding

the spatial attribute.

$$acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1.1}$$

$$bacc = \frac{(Precision + Recall)}{2} = \left(\frac{TP}{TP + FP} + \frac{TN}{TN + FN}\right) \cdot 0.5 \tag{3.1.2}$$

TP: True Positives, TN: True Negatives, FP: False Positives, FN: False Negatives

Both of those metrics have a different meaning and tell us about different risk-related aspects of the data set. While the accuracy represents the actual number of individuals at potentially higher risk, because their spatial identifier was re-identified, the balanced accuracy weighs every class equally. Consequently, the re-identified individuals from smaller classes contribute more to a high balanced accuracy value, which is desirable in our case because the re-identification of individuals from smaller classes provides a higher increase of information for the attacker.

3.2 Population Uniqueness

The Population Uniqueness [T K23] Ξ_t for a data set D_{syn} is defined as the proportion of records i being unique among a certain subset D^t_{syn} with $t \in [1, ..., m]$ attributes, while the respective record i in D^t_{orig} is unique as well and additionally $D^t_{orig,i} = D^t_{syn,i}$ applies. This proportion naturally increases the more attributes are added to the subset, because more and more records end up being unique, while at the same time the proportion also decreases, because it gets less likely that all attribute values of the synthetic and original records are identical. The authors choose the approach of adding the attributes in a random order to the subset D^t_{syn}, which however stays the same for the examination of the different data sets. The resulting Population Uniqueness Ξ_t tells us what proportion of individuals could theoretically be uniquely re-identified if the attacker exactly knew those attribute values of the given subset D^t_{true}. Thus, it gives us information on the risk of identity disclosure, which automatically also influences the risk attribute disclosure.

4 Utility

One of the main objectives of data synthesis is to generate data sets that are altered, yet still useful for various analyses and for obtaining reliable results by the user. For obtaining quantifiable metrics on this usefulness of the synthetic data we will perform an evaluation of utility. Since we trained a different synthesis model for every stratum, we will also calculate the utility evaluation metrics separately for every stratum and accumulate the results.

[DH23] discusses various strategies for quantifying the utility of the generated data and divides them into three categories. The first one are the so-called global utility metrics, which are usually computed by directly comparing the synthetic and the true data. They offer the huge advantage that no assumptions need to be made regarding the analyses carried out on the synthetic data. However, they only give an idea of the general utility potential and are not necessarily transferable to a specific analysis the user might be interested in. For those cases, outcome-specific utility metrics can be used, which tell us about the utility for one specific analysis. They can be seen as use cases, for instance for the estimation of a certain distributional parameter, for which one would compare the estimations based on synthetic and original data with each other. The third category are the fit-for-purpose measures. They can be used for data exploration at the beginning of any utility assessment. Those could be graphical comparisons of marginal or bivariate distributions on similarity, as well as consistency checks for implausible values in the synthetic data.

For this project, we want to get a broad overview over the potential for allowing a reliable comparison of the different synthesis methods, which is why we choose at least one method from each of the three categories.

4.1 Fit-For-Purpose Metrics

For quantifying how well the distribution of the synthetic data matches the original distribution, we are searching for methods that can be applied to our mainly categorical data set. We will apply those to the marginal and bivariate distributions, in order to also learn how well the synthesis preserves the relationship between the different variables.

The authors of [RND21] have analyzed various Fit-For-Purpose measures that are calculated by cross-tabulation. For the application to numerical attributes, these attributes need to be grouped in advance. The authors calculated ten metrics empirically for 120 syntheses and found that almost all the metrics were highly correlated, in most cases even $\rho \geq 0.99$. The only metric that did not show a correlation of $\rho > 0.9$ constantly with every other metric was the likelihood-ratio test statistic. In order to obtain a profound idea of the utility regarding univariate and bivariate distributions we will calculate the likelihood-ratio test statistic and one other metric that is highly correlated with many other metrics. As other metric we will choose the Pearson χ^2 Statistic. This choice is made because it is highly correlated ($\rho \geq 0.99$)

with the Freeman-Turkey statistic, the Jensen-Shannon divergence and the pMSE *(Propensity Score Mean Squared Error)*, the latter of which we will use later in section 4.2. Moreover, the Pearson χ^2 Statistic is a well-established and commonly used metric, also outside the field of data synthesis.

In the following formulas for the test statistics, $l \in L$ are all the levels of the respective attributes of D_{syn}, while n_{syn} are the counts of class l in the synthetic attribute and n_{orig} the counts of class l in the original attribute. For multivariate tests, where the distributions of two or more attributes are compared, $l \in L$ stands for all possible tuples of classes of the examined attributes. For instance for a bivariate test where attribute a has 4 classes and attribute b has 3 classes, the resulting term would have 12 summands.

We want to point out that the test statistics calculated here do not act as hypothesis tests, because the synthetic data is generated from the distribution of the original data, which means that the data sets cannot be considered independent samples. Therefore, the p-value of the tests cannot be used to make assumptions about whether the two distributions are statistically equal or not, like in a χ^2-Homogeneity test. However, we can compare the test statistics of the different data sets to find out how close the distributions of D_{syn} and D_{orig} are. Low Test-Statistic values would mean that D_{syn} and D_{orig} have similar distributions, which would indicate a high level of utility of the synthetic data, while high Test-Statistic values mean that the distributions of both data sets significantly differ, which indicates a low level of utility.

4.1.1 VW-Test (Pearson Chi-Squared Statistic)

The Test-Statistic for the adjusted χ^2-Statistic as in [RND21] for utility evaluation can be found below in equation 4.1.1. The VW-Test Statistic includes a little difference to the commonly known Pearson's χ^2-Test Statistic for homogeneity. [VW01] proposes to replace the denominator of the formula with the mean value of the original and synthetic counts to avoid division-by-zero errors, in case the new synthetic data falls into groups that are not present in the original data. Even though our applied synthesis algorithms prevent such cases, we will calculate the VW-Test Statistic in this commonly used way. The VW-Test as in [RND21] is implemented in R as synthpop::utility.tables(tab.stats='VW') [NRD16].

$$T = 2 \cdot \sum_{l \in L} \frac{(n_{syn} - n_{orig})^2}{(n_{orig} + n_{syn})} \tag{4.1.1}$$

4.1.2 G-Test (Likelihood Ratio Chi-Squared Statistic)

The calculation of the Likelihood Ratio χ^2-Statistic as in [RND21] is also referred to as G-Test. The respective formula is displayed in equation 4.1.2. The authors add that the G-Test is not suitable for sparse tables, which is the consequence of the logarithm not being defined for the case $n_{syn} = 0$. Therefore, only levels where both $n_{syn} > 0$ and $n_{orig} > 0$ apply are used for the calculation. For sparse tables that would mean that many attributes that are actually present in the data are not part of the metric. Since in our case we only use tables based on a maximum number of two attributes while having data frames with

high numbers of records, the resulting cross-tables and therefore our calculations are not expected to be affected by those issues. The G-Test as in [RND21] is implemented in R as synthpop::utility.tables(tab.stats='G') [NRD16].

$$G = 2 \cdot \sum_{l \in L} n_{syn} \cdot \ln\left(\frac{n_{syn}}{n_{orig}}\right) \tag{4.1.2}$$

4.2 Global Utility Metrics

For quantifying the Global Utility of the synthetic data set, a method commonly used according to [DH23] is the usage of propensity scores from a model that tries to discriminate between synthetic and true data records, based on supervised learning. Propensity score methods are based on the work of [RR83] on propensity score matching. [RM09] were the first to introduce them as a measure to evaluate global utility of synthetic data. There is an existing implementation in the synthpop-package in R, as synthpop::utility.gen() [NRD16].

As explained in [Sno+18], for this method D_{syn} and D_{orig} are stacked to create D_{comb}. Now an additional binary variable b gets added to D_{comb}, which indicates whether a record is from D_{syn} or D_{orig}. Then a model is fitted, using D^{b}_{comb} as target and D^{-b}_{comb} as predictors. From that model, we obtain a vector with the propensity scores $\hat{p}$ representing the predicted likelihood of $i \in D_{syn}$ for every record i.

The authors of [Sno+18] also suggest several models for estimating the propensity scores. While for simple synthetic data sets they recommend logistic regression models with first-order interaction terms; for more complex data sets they recommend to include higher order interaction terms as well. For big data sets with a large number of attributes, as in our case, they recommend using CART models, because parametric models with higher order interaction terms would probably not be computationally feasible anymore. Another argument for using a CART model is that there is no need to first encode our categorical variables.

After calculation of the propensity scores $\hat{p}$ with a CART Model, we need to find a way to calculate a meaningful utility metric based on them. According to [DH23], the pMSE (equation 4.2.1) is a currently popular metric, which was introduced by [MK09] for this purpose, together with the propensity score calculations.

$$pMSE = \frac{1}{N} \sum_{i=1}^{n} (\hat{p}_i - c)^2 \tag{4.2.1}$$

The difference to the regular Mean Squared Error is that c stands for the value of p_i under a perfect synthesis. In this case, the propensity model would not be able to distinguish between records from D_{syn} and D_{orig} at all, and constantly predict the proportion of synthetic records among D_{comb}, which is 50%. For our case we will therefore use $c = 0.5$. It is important to mention that the pMSE is highly dependent on the original data source and the power of the underlying discrimination model for the propensity scores. Moreover, [MK09] adds that its value naturally increases with the number of predictors in the data set. Thus, this value can

only be used to compare synthetic data sets from the same source with equally calculated propensity scores with each other, not to draw general conclusions about their utility.

4.3 Outcome-Specific Utility Metrics

Besides metrics that compare the distributions of the different data sets, we are also interested in how well an actual analysis on the synthetic data would perform. In our use case, we calculate the unemployment rate for synthetic and original data and compare the results. Therefore, we will assume the binomial and identically distributed variables $\mathcal{X}_i$, which tell us whether individual i is unemployed or not. We will calculate the unemployment rate r_{syn} for sub-stratum levels on D_{syn} and compare it to the same analysis r_{orig} on D_{orig}.

For comparing the results, we will calculate two metrics. Firstly, we calculate the relative error of the estimated rate, which we calculate as $\frac{|r_{orig}-r_{syn}|}{r_{orig}}$. In case $r_{orig} = 0$, while $r_{syn} \neq 0$ applies, in order to allow a finite aggregation, we will calculate $\frac{|r_{orig}-r_{syn}|+1}{r_{orig}+1} = r_{syn} + 1$. Secondly, we calculate the CIs *(Confidence Intervals)* Overlap CI_{over} of the 95% CI, as mentioned in [DH23]. The CI Overlap represents the percentage that the confidence intervals for the unemployment rates obtained from the original and synthetic data overlap. Since we assumed $\mathcal{X}_i$ to be binomial and identically distributed, the 95% CIs are calculated as displayed in equation 4.3.1.

$$[\mu - z_{95\%} \cdot \sigma, \mu + z_{95\%} \cdot \sigma] \tag{4.3.1}$$

With expectation value μ and the standard deviation being

$$\mu = n \cdot p, \sigma = \sqrt{n \cdot p \cdot (1-p)} \tag{4.3.2}$$

and $z_{95\%}$ being the z-score for the 95% CI. We first calculate the CIs, using the R function stats::prop.test(). Based on that, we calculate CI_{over}, which is how many percent of the unification of their CIs both intervals share. If both intervals do not intersect at all, or one of the data sets does not include any individuals within the respective area, we will use $CI_{over} = 0$.

5 Empirical Examination

5.1 Data Preparation

In this section, we describe the preparation of our data set. The preparation was carried out in a way so that it could be processed by the synthesis algorithms. This includes creating the raw data table by joining the provided sub-tables, data cleaning and finally the treatment of missing values, which also must be considered in depending on the synthesis models used later on.

5.1.1 Acquisition of Raw Data

Due to its huge amount of data with complex relations, the Census Data is split up into many sub-tables that can be joined via matching keys. First, we had to join the sub-tables to the dataset required for our project, directly discarding unused attributes. The decision which attributes to keep and which ones to discard was made based on the following criteria. There should be a potential interest in synthesizing this specific attribute because a potential attacker could either be interested in the attribute values or use them to reveal the values of other attributes. Keys and IDs would therefore not need to get synthesized, since their value has no semantic meaning anyway. Moreover, IDs for linking related individuals with each other, such as parents or children, were ignored in this project. Furthermore, duplicate or almost duplicate variables were discarded. These often occurred, if categorical attributes had a national and an EU-based, or additionally even another sub-national representation. Similarly, in the relationship between the two attributes for age and year of birth the two attributes make each other one almost completely redundant. After removing all attributes that were not to be used, the final data set consists of 26 attributes, including PERSON_ID, the unique identifier. The summary of this data set can be found in table 7.1.

5.1.2 Data Cleaning

Generally, the provided data tables were already in a well-kept condition and only a few minor data cleaning steps had to be applied. One aspect was that many variables that could actually be stored and presented as factor (R-specific format) were stored in a character (string) format, which meant an unnecessary big usage of memory. This was due to the data being provided in a CSV format, which does not support factors. Since how ever R and RDS (R Data Sets) do so, those values were converted to this less memory-consuming format. Only the unique identifier PERSON_ID was converted to integer instead of to factor. In the summary of the final data set in the appendix 7.1 we can see that there are 20 factor attributes and five integer attributes (without PERSON_ID), almost all of them with 11 or less unique values, but no variable in double (float) value format.

In order to detect illogical or corrupted values, the unique values for each attribute were viewed. Only for the age variable ALTER_01JS, a summary of the distribution was viewed instead. As we found, there were hardly any illogical or unexpected values. Besides some missing values, all attributes had exclusively values in agreement with the provided value glossary of the meta data [Bay16], except for the geospatial identifiers. The geospatial attributes, except for the AGS_12, had about 10K records with an obviously corrupted character string, which at this point were all set to NULL. The background and treatment of the missing values will be discussed in the following section 5.1.3.

5.1.3 Treatment of Missing Values

Besides the general data cleaning steps, like checking for and treating illogical or corrupted values, one essential step is to treat the missing values before the synthesis for every synthesis model, according to the specific requirements of the model. The original data tables did not come with any missing values among the semantic attributes (non-key attributes). However, due to some missing or corrupted key values, the joins resulted in two different groups of missing values in the raw data set.

The first group occurred because of some foreign keys (GEBÄUDE_ID, WOHNUNGS_ID) in the table PERSON (which contains one record per individual) having missing IDs, which prohibited links with the tables for housing and buildings. About 1.54M records were affected and therefore had missing values in six of the 26 attributes of the final data frame (details in table 7.1). The missing keys occurred for individuals for which no connection to any residential object could be established [Sta16]. Accordingly, these are common incidences and result from missing or insufficient residential capacities at the respective property or address. Presumably, those individuals did either provide an obviously incorrect address or no address at all. The resulting missing values carry a semantical meaning and can therefore hardly be considered MCAR *(Missing Completely at Random)*. At the same time, other attributes, like the status of an individual's family (FAMSTAND_AUSF), are expected to have some dependency on the affected attributes, which is why assuming MAR *(Missing at Random)* seems appropriate.

To treat this first group of missing values in the six affected attributes, we need to distinguish between categorical and numerical attributes. For categorical attributes, the missing values can be introduced as a separate class, as long as they carry a specific semantic meaning, which is the case here. For the numerical attributes, a possible way of treatment would be imputation. Unfortunately, the linear imputation with its relatively low computation costs cannot be used on mixed data sets consisting of categorical and numerical attributes. Theoretically, we could use One-Hot Encoding on the categorical attributes to solve this problem, but this would lead to the following issues: First of all, the data set already consists of over 80M observations. Using such an encoder on all of the 20 categorical attributes, of which many of them have ten classes or more, would make the amount of data too large to handle. Moreover, the geographical attributes with information on county and coordinates have so many classes that One-Hot Encoding would be impossible. The respective attributes would have to be excluded from the imputing procedure. Other imputation algorithms can actually handle mixed data sets, for instance, one by [AHJ13] that uses PCA *(Principal Component Analysis)* to handle the huge amount of attributes resulting from the One-Hot Encoding and

is implemented in R as missMDA::imputeFAMD(). However, this imputation method would be computationally also extremely expensive on our data. Theoretically, we could simply choose only a few attributes that prove to have a dependency on the attributes affected by missing values. A possible choice would be FAMSTAND_AUSF. Unfortunately, this attribute is categorical. Due to the high number of classes in FAMSTAND_AUSF, using the mentioned imputation method would still lead to a very expensive calculation. Moreover, the calculation of a correlation coefficient for examining whether there is an actual dependency between the variables of different data types, would hardly be possible within a reasonable amount of effort. Another aspect to consider is that the objective of this project is to investigate methods to synthesize the true census data. Imputing missing values would always create data further away from reality than the original data. Taking this last aspect and the potential problems imputation could cause into account, it seems reasonable to try and find a way to make the synthesis models work despite the remaining missing values in the numeric attributes.

In the following, we will explain the functionalities of the different algorithms in processing the data. Geomasking relocates the spatial identifier of an observation, which affects attributes with geographical information while all other attributes remain untouched. Since the missing values occur exclusively among non-geographical attributes, the attributes affected by missing values can be left as they are. The Random Forest Synthesizer and the CART Synthesizer both use classification trees. Thus, before running this algorithm on our data we have to transform the integer attributes to categorical attributes anyway. This means that we can introduce the missing values as a separate class before running the synthesis. More details on the transformation from integer to categorical attributes will be presented later in section 5.1.4.

Both Copula Syntheses models only run on numerical data, which is why all of the categorical attributes are frequency encoded before the multivariate density functions are estimated. On the encoded data set, the only calculations carried out are the attribute-wise estimation of the means and variances, as well as the pairwise estimation of the covariances. Let us first take a look at the mean estimators. If the missing values x_j^{NULL} were to be ignored, the resulting estimators for μ_j would be biased, because $mean(x_j^{-NULL}) = mean(x_j^{NULL})$ generally does not apply for MAR attributes. However, we could consider the calculated estimator $\mu_j = mean(x_j^{-NULL})$ as only applicable for x_j^{-NULL}. Thus, we are allowed to sample from a distribution with this estimator, as long as we set the previously missing values as missing again after synthesis. In addition to μ_j, we want to estimate covariance estimators ρ_{j_1,j_2} without any bias. If we use the setting *use = "pairwise.complete.obs"* in the R function base::cov(), we calculate the covariance estimator between two columns only based on value pairs where both values are not missing. If we assume $cov(x_{j_1}^{-NULL}, x_{j_2}^{-NULL}) = cov(x_{j_1}, x_{j_2})$, we can consider ρ_{j_1,j_2} as applicable for the entire attribute. We decide to make this assumption for enabling the calculations for the synthesis algorithms; moreover, the same assumption would have to be made for using Linear Imputation. Since Missing Value Imputation turns out to be neither convenient nor necessary for running our synthesis algorithms, we simply decide to leave those missing values as they are.

The second group of missing values affects about 10K records of individuals from the table PERSON, which came with missing values among almost all the spatial attributes. As mentioned in section 5.1.2, the attribute AGS_12 was the only geographical attribute not to be affected by missing values. AGS_12 is the commune-level spatial identifier and is a

composition of keys from different administrative disaggregation levels. Therefore, it was possible to figure out that the corresponding regional identifier 1700 from REGION_KREIS on does not exist. Those missing values have to be considered as NMAR *(Not Missing at Random)*, because all of the affected records are from a specific county and there is no reasonable way to deduct from the available attributes the missing geographical identifier. While in other data science problems missing values of type NMAR might cause bigger issues, in this case we estimate a separate model for each county. Consequently, those missing values could only have an effect on the county where the records are actually located. Within the affected county or counties there is no way of knowing what type of missing data the missing attributes have. Also, we do not know which of the 412 counties those records are located in. Consequently, there is no alternative to removing the affected variables from the data set.

5.1.4 Treatment and Grouping of Integer Attributes for Methods requiring Categorical Attributes

For preparing our data set for the CART and the Random Forest Synthesizer, it is necessary to transform the integer variables to categorical variables. The same applies for some of the algorithms for risk and utility metrics, such as the discriminator on propensity scores, both χ^2-distributed statistics and the privacy attack algorithm. For attributes with low numbers of classes this is possible without any further adjustments and even solves the issue of missing values, as explained in section 5.1.3. For integer variables with higher numbers of levels, which only applies to ALTER_01JS, we will apply grouping as follows: We simply round the attribute values to tens $(0, 10, 20...)$, which results in eleven groups. This comes with the advantage of not having to check any constraints on the synthetic samples to obtain logical value levels, since decision trees will only predict classes present in the original data. A re-transformation of this grouping is not necessary at any point.

5.1.5 Treatment of the Singular Spatial Attribute for Copula, CART and Random Forest Synthesis

The stratum 9461 only has one value for the attribute GITTER_ID_10KM, which for the Copula Synthesis makes the required estimated covariance matrix singular. Similarly, the algorithm for Classification Trees, used for CART and Random Forest Synthesis, does not support predictors with only one class. Since the affected county has a relatively small population $< 70k$ anyway, the main concern was to find a fix to make the algorithm run for all synthesis models without producing errors. Therefore, we introduced two artificial factor levels for the attribute GITTER_ID_10KM and randomly split the population of this stratum into two equally sized groups. One of the new levels was assigned to the first group, the other one to the second group.

5.2 Steps and considerations to permit results comparable with [T K23]

For assessing the potential of the data synthesis methods, we calculate various metrics for quantifying the utility and re-identification risk. We would like to compare our findings for the Geomasking and the Copula Synthesis with Frequency Encoding to the findings from [T K23] using these metrics. Thus, we want to apply these synthesis methods to our data set in a way as similarly as possible to [T K23]. In this section, we will describe our thoughts and considerations that led us to applying the synthesis algorithms to our data set the way we did.

5.2.1 Costa Rican Census Data and Administrative Disaggregation Structure

[T K23] apply the Copula Method and the Geomasking to a 10% sample of the 2011 Costa Rican Census, which contains data of the entire population of Costa Rica. The sample consists of 427,830 records and 106 attributes and can be accessed via [SC]. In order to understand how [T K23] apply the synthesis methods to their data set, we have to take a brief look at the administrative structure of Costa Rica.

Administratively, Costa Rica has among other disaggregation levels six planning regions, 81 cantons, and 473 municipalities. The municipalities represent the smallest geographical information available in the Costa Rican Census. Their identifier is the zip code. The zip code is also the most granular spatial information available in their data set.

5.2.2 Stratification and application of the algorithm to the data

How [T K23] applies the algorithm to the data

In the paper, the authors used the same strata as in the sampling design for the Main National Household Survey ENAHO *(Encuesta Nacional de Hogares)*, where twelve strata are used, as each of the six planning regions is further disaggregated into urban and rural municipalities.

For the Copula Method, [T K23] estimated a separate synthesis model for each of the strata. The authors use the zip code as spatial identifier for the conditional sampling.

For the Geomasking, the authors sampled the distances d for moving the locations of the individuals from uniform distributions depending on whether the respective municipality is rural or urban. For rural municipalities, they sampled the distances for the relocation (in meters) as $d \sim U[0, 5,000]$, for urban municipalities, they sampled the distances as $d \sim U[0, 2,000]$. After moving the location, the condition was that the new location still should be within the same canton, otherwise the new location would be rejected and a new sample would be drawn. Using this method for obfuscating the true location, only roughly 30% of the records were assigned to a new zip code, while the other 70% remained with the same zip code.

How we apply the Copula Synthesis with Frequency Encoding to the German Census data

For the application to the German Census Data, our objective was to find a sufficiently low-level disaggregation that would provide strata for allowing precise modeling of the local population structure, while still including a solid number of individuals in every stratum and thus keeping disclosure risks low. A too little number of individuals in one stratum would cause the synthesis model to be overfitted. Moreover, the synthetic samples would be more likely to reveal too much information about individuals from the underlying data.

As in the paper [T K23], we want to base our stratification on administrative areas. Consequently, we will select one level of administrative disaggregation areas and use them as strata for our project. Table 5.1 displays the levels of administrative disaggregation we could use for that purpose.

Table 5.1: Administrative Disaggregations Germany

Disaggregation	Official Name	Variable	Unique Values
Square 100x100m	-	GITTER_ID_100M	3,296,697
Square 1x1km	-	GITTER_ID_1KM	217,992
Square 10x10km	-	GITTER_ID_10KM	3,826
Commune,City	Gemeinde,Stadt	AGS_12	11,491
County,City	Landkreis, Kreisfreie Stadt	REGION_KREIS	412
State	Bundesland	- not in data set-	16

A general challenge is the strongly varying numbers of individuals within different administrative disaggregation levels. Considering the disclosure risk, we decide to view the number of individuals per administrative area on each level (table 5.2) and choose the most granular disaggregation level where the lowest numbers of individuals are still sufficiently high.

Table 5.2: Population per Administrative Area in each Disaggregation

Disaggregation	Minimum	Maximum	Mean	Median
Square 100x100m	1	1,969	24.3	13
Square 1x1km	1	23,379	368	59
Square 10x10km	1	850,247	20,964	9,580
Commune,City	11	3,292,365	7,074	1,666
County,City	34,200	3,292,365	194,684	137,069

As we can see, only on county-level we have minimum population numbers that are consistently high. Therefore, we choose the counties for the stratification.

When running the synthesis algorithms later on for both Copula Syntheses, we would now draw synthetic samples, conditioned on a low-level spatial identifier. Similarly, the CART and Random Forest Synthesizer would use this identifier as predictor. Thus, we need to choose another disaggregation level below strata-level for that. On one hand, we would like this disaggregation to be granular, in order to generate accurate synthetic data for precise spatial identifiers. On the other hand, all synthesis models come with upper limitations for the number of classes that still would be processable. For the Copula Synthesis with Frequency Encoding, this limitation is the number of classes that can reasonably be represented in a single frequency encoded vector. The more classes there are, the less accurate the conditional samples will describe the population structure in the sub-stratum areas. For the CART and Random Forest Synthesizer, it is the maximum number of classes that one predictor can have. In the R function RandomForestSRC::rfsrc(), this number is 53, although some workarounds (splitting up an attribute into multiple attributes e.g.) could also allow processing variables with more classes. In order to choose our sub-stratum disaggregation level, we decide to view the number of those sub-stratum areas per stratum for each sub-stratum disaggregation level.

Table 5.3: Number of disaggregation units below stratum level

Disaggregation	Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum
Square 100x100m	920	4,726	7,154	8,006	10,310	40,247
Square 1x1km	34	231	496	541	756	2,077
Square 10x10km	1	8	16	16.6	23	48
Commune,City	1	1	21	27.5	37	235

Table 5.3 shows us that only the population counts for the 10x10km grid cells and maybe for the communes are sufficiently low for allowing the synthesis models to function properly. In order to decide between those two options, we take a look at the lower 1st quartile of the distribution. We can see that many of the strata include only one commune because the underlying attribute AGS_12 does not further disaggregate within most bigger cities. This implies that in cities, like Berlin, Hamburg, and many others, we would completely waste the potential of our synthesis models to consider sub-stratum level differences in the population structure. Since all communes and cities, with one exception, are big enough to include more than one 10x10km grid cell, we will choose the attribute 10x10km_GITTER_ID as sub-stratum disaggregation level and therefore as spatial identifier *loc*.

5.2.3 Reasons to refrain from sampling from our data set

The sampling process in [T K23] is a stratified two-stage cluster design. They introduce the PSUs *(Primary Sampling Units)* based on the municipalities. Due to the fact that single municipalities can include both rural and urban areas, some of them were disaggregated further, which resulted in 767 PSUs. For empiric reasons, this sampling process was repeated 100 times and the risk and utility examinations were applied to each of the 100 synthetic data sets. Their sampling method consists of two steps. In the first sampling stage, the authors sampled on average 123 out of the 767 PSUs, separately for each stratum. The selection probability for each PSU was proportional to the population size of the respective stratum.

In the second stage, they used random sampling without replacement to sample records from all selected PSU. Afterwards, they discard PSU with less than 10 selected individuals from the procedure, which on average affected around 4% of PSU. The resulting data sets contain $n = 7.638 - 11.914$ records and holds about 2% of the original 10%-sample.

We need to keep in mind that in this project, our goal is to answer the question how well the examined synthesis algorithms would perform on the German census data or on similar data sets with spatial attributes. Therefore, we want to use a sampling method that will not affect the performance of the examined synthesis algorithms positively or negatively.

As described above, the authors of [T K23] perform the data synthesis only on a 2% sample of the originally available data set, which moreover only contains data from on average 123 of all 767 existing PSUs. Our question is, whether the synthesizing abilities of the examined algorithms are affected, if on average only ten instead of 64 PSUs are used per stratum. In that case, the encoded original data set $\tilde{D}_{orig}$ would have a frequency encoded vector with on average only 16% of the actual unique values, of which all of them are supposed to represent a different conditional distribution. Details to the calculated numbers can be found in the equations 7.0.1 and 7.0.3. According to our considerations, it might be possible that the performance of the conditional sampling might not be as good when all PSUs are used instead. This would mean that in reality, when the synthesis is performed on the entire data set, the utility of the low-level spatial synthetic data from [T K23] might be worse.

After the second sampling step, [T K23] remains with on average 70 individuals per PSU, compared to the original 558 individuals per PSU that would be present if the complete data set was used. Our question is whether this difference could also cause significantly different results for the utility and privacy examination. To give an example: If, due to sampling, one stratum only had a very small number of individuals, the risk evaluation of the synthetic data might suggest some increased re-identification risk for individuals within the respective stratum. However, in reality this risk might not exist because the entire data set with a much higher number of individuals would be synthesized. Moreover, one could argue that due to the much higher number of records, in reality the parameter estimation might perform better, which again could indicate that the algorithm would perform better on the full data set.

Considering the potential shortcomings that come along with examining the performance of the synthesizers on samples, we have decided to use all data records from every stratum in this project. We only perform one run for each model and parameter setting due to computational reasons, unlike the 100 runs in [T K23]. This significantly reduces empirical power, but on the other hand, we do not run into any of the sampling-caused problems mentioned before. By using this conservative sampling strategy (which for the synthesis means no sampling at all), we accept a slightly higher variance among our key values for describing the utility and re-identification risk, while avoiding different biases of unknown extent.

5.3 Shape Files

During the Geomasking, one part of the algorithm is to check whether the new location of an individual is within the same stratum as before. Thus, a shape file with all of the county

borders of Germany is required, for which we use the files [Kar] for. They include a layer of the county polygons and the necessary metadata, which we both combined to a geospatial data frame.

As preparation for the algorithm, we preprocessed the geospatial data as follows. Since some counties consist of two or more polygons, all polygons of one county were merged to one multi-polygon. We obtained the number of inhabitants of every stratum from the census data by counting the number of records for each stratum to attain the population density of the counties. Then, the geospatial multi-polygons were merged with the census data, using the county ID REGION_KREIS. The population density was calculated by division with the polygon size, which was also an attribute of the geospatial data. Now that the multi-polygons and the population density were added to the original data as additional attributes, the spatial identifier GITTER_ID_100m was transformed to geospatial coordinates. Since the Geodetic CRS *(Coordinate Reference System)* that the spatial identifier GITTER_ID_100m was based on is the Lambert Projektion LAEA (EPSG:3035), they first had to be transformed to UTM zone 32N (ESPG:25832), the CRS of the multi-polygons. After the Geomasking algorithm was run, as described in section 2.2.5, the new coordinates were transformed back to their original coordinate system and matched with their respective spatial identifier from GITTER_ID_100m.

5.4 Hyperparameters

For the different synthesis models, we have the option to choose different hyperparameters. We are interested in finding the best hyperparameter combination to achieve the best utility-risk ratio possible. Hyperparameter optimization can get computationally expensive, since usually the model needs to run for every possible combination of hyperparameters within a set range, if possible for the entire data set. Hence, we need to select the hyperparameters we want to check, their values we are interested in, and the sample size of the data set to test on.

5.4.1 Choice of Parameters, Parameter Values, and Data Sample

Looking at how we will be applying the synthesis algorithms, we find that we apply the same algorithm for every stratum. Thus, it seems reasonable to just randomly sample some of the strata and perform the optimization on them, instead of having to calculate the metrics of interest for the entire data set. We decide to choose 40 randomly sampled strata, which includes slightly below 10% of the entire 412 strata. Which hyperparameters could theoretically be optimized can be found in table 5.4.

For the Geomasking we use the only possible parameter n_{closer}, which is the number of individuals who are expected to be closer to the true location of individual i than the synthetic location of i is. We choose to optimize for four different values of n_{closer}, as to be found in table 5.5.

For the CART and Random Forest Synthesizer, there is a wider choice of hyperparameters. Regarding the synthesis order of the variables, we will stick with ordering them in ascending

Table 5.4: Hyperparameters

	Copula Synthesizer Frequency Enc.	Copula Synthesizer One-Hot Enc.	Geomasking	CART Synthesizer	Random Forest Synthesizer
Available	-	-	N_{closer}	Parameters for pruning, Synthesis order of variables	Parameters of the Random Forests, Synthesis order of variables
Chosen	-	-	N_{closer}	Maximum number of records in final node n_{mtry}	Number of trees n_{trees}, Minimum nodes size n_{mtry}, Sample size n_{samp}

order by number of unique values, which is due to computational reasons and was already pointed out in section 2.2.3 and section 2.2.4. Regarding the Classification Trees and Random Forests, we choose parameters that influence how precisely the original data will be learned, and therefore, how precisely the synthetic data values can be predicted. If the prediction is too precise, the re-identification risk might be too high, while too low precision of the models can make the synthetic data useless for precise geospatial analysis.

For the CART Synthesizer, we will only optimize one parameter for regularization, which is the minimum allowed number of observations in any node of the tree n_{mtry}. Since this synthesis method has relatively low computational costs, we will optimize using five different values. The selected parameters for the Random Forest Synthesizer are the number of trees in every forest n_{trees}, n_{mtry} as for the CART Synthesizer and the number of records n_{sampl}, which is selected from the N records in D_{true}. We chose three different values for each hyperparameter for the grid search (see table 5.5) because the grid would increase drastically with any additional value. Even though we might not find the absolutely optimal values for each hyperparameter in that way, we will still get an idea of which of the hyperparameters might have an influence on the synthesis, and which not.

Table 5.5: Hyperparameter Values

Geomasking	n_{closer}	4,000;16,000;64,000;256,000
CART Synthesizer	n_{mtry}	1;3;5;10;15
Random Forest Synthesizer	n_{trees}	50;200;500
	n_{mtry}	1;3;10
	n_{samp}	$N^{0.6}, N^{0.7}, N^{0.8}$

5.4.2 Choice of Evaluation Metric

For synthetic data, there is not just one criterion to be optimized, like a loss function, but there is a compromise between utility and risk to be made. For our evaluation, we will therefore have to consider utility and risk metrics and choose those hyperparameters that lead to the overall best results, according to our judgment. It would be desirable to use metrics that are calculated based on the entire synthetic data set and return us only one final value as a result, instead of for instance one value per attribute. The latter would force us to think of a way to accumulate those values properly into a meaningful result that would allow comparison between the different hyperparameters. For the same reason, we only want to use one risk and one utility metric each. Among the metrics presented in the chapters 3 and 4, there is only one metric each that delivers one final figure for describing the entire data set. As evaluation metrics we will therefore use the Propensity Score Model (section 4.2) for utility and the Privacy Attack (section 3.1) for risk evaluation.

5.4.3 Results of Hyperparameter Optimization

For the Random Forest Synthesizer, the hyperparameters generally do not seem to have a severe influence on the risk and utility of the synthetic data, as to be found in figure 5.1. More precisely, for the examined parameter values the only parameter that might have some influence is the number of trees n_{trees}. This influence mainly affects the pMSE only. The pMSE is lowest for $n_{trees} = 200$, which is why this parameter is chosen. The slightly lower accuracy values for $n_{trees} = 500$ do not influence our choice, also because Random Forest Synthesizers tend to have a low re-identification risk anyway. Since no influence of the other two parameters n_{mtry} and n_{samp} can be found, we choose the values expected to result in the cheapest calculation, which are $n_{mtry} = 10$ and $n_{samp} = N^{0.6}$.

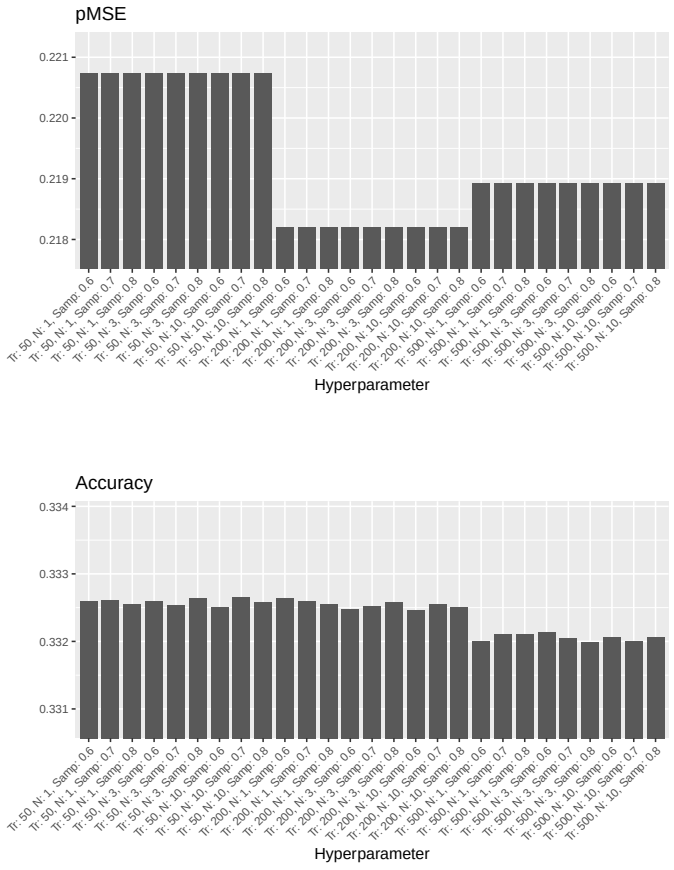


Figure 5.1: Hyperparameters for Random Forest Synthesizer

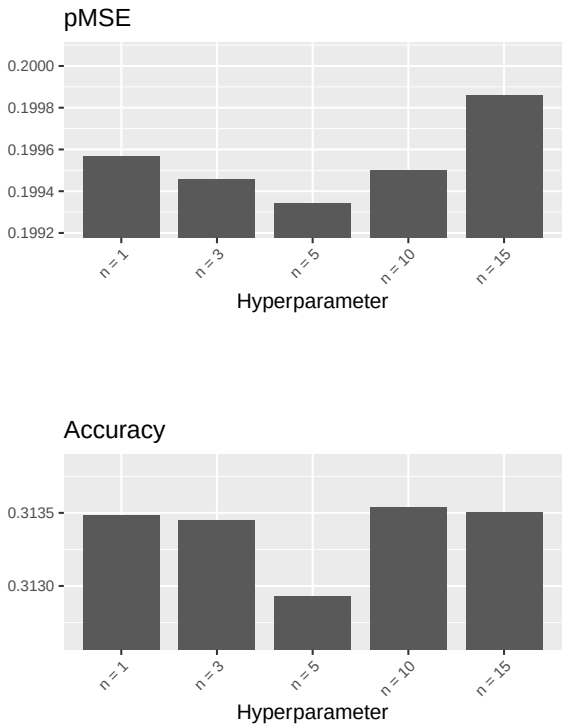


Figure 5.2: Hyperparameters for CART Synthesizer

For the CART Synthesizer, the influence of the examined hyperparameter n_{mtry} is also insignificant. Examining the accuracy results, we expect a negative correlation between n_{mtry} and the accuracy, since a tree with fewer nodes is expected to predict D_{syn} less close to D_{orig}. Therefore, it is surprising that $n_{mtry} = 5$ produces the lowest accuracy, although the relative differences between the different values are negligible. Since we want the CART Synthesizer to primarily focus on producing good results for the utility metrics, we again make our choice based on the pMSE. We will therefore use $n_{mtry} = 5$, which [Rei05] also recommends as default value for data synthesis.

For the Geomasking, we find that both the pMSE and the accuracy are lowest for the biggest value $N_{closer} = 256,000$, which means that this would have to be the value of our choice. However, we have to consider that the results of the discriminator for the propensity scores might not be perfectly suitable for examining this synthesis method, since only the attribute values of GITTER_ID_10KM are altered. The discriminator, however, considers the distribution of all attributes. Another hint that the pMSE might not be a perfectly valid metric for the Geomasking is that the pMSE and the accuracy seem to be positively correlated. In theory, a better utility and thus a lower value for the pMSE should come with a higher re-identification risk, thus a higher accuracy. Consequently, we decide not to consider the pMSE from the propensity scores for choosing the hyperparameter values. When looking at the different accuracies of the Privacy Attack we notice that there is a strong decrease between $N_{closer} = 16,000$ and $N_{closer} = 64,000$, the decrease between $N_{closer} = 64,000$ and $N_{closer} = 256,000$ however is only minor. If we take a look at the actual distances d that the locations of the individuals were on average moved during Geomasking, as presented in table

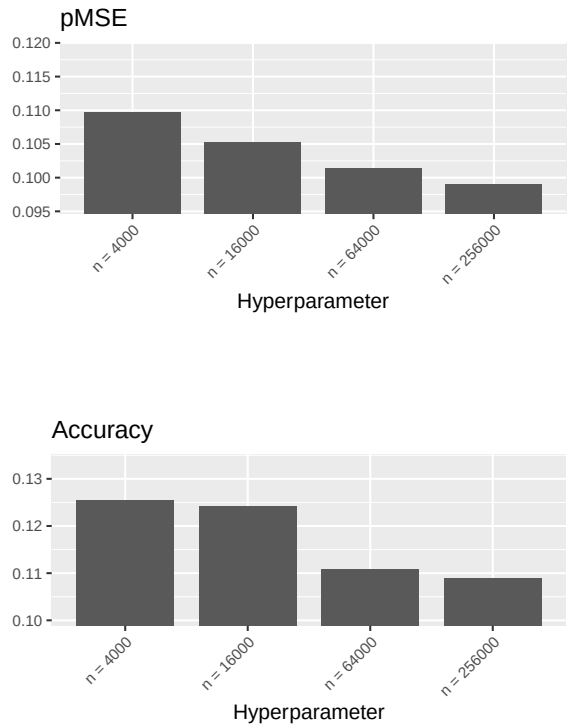


Figure 5.3: Hyperparameters for Geomasking

5.6, we see that between $N_{closer} = 64,000$ and $N_{closer} = 256,000$ there is still a difference of 29.5%. This can make a severe difference for the utility of synthetic data for low-level spatial analysis. To sum up, we choose $N_{closer} = 64,000$ as hyperparameter for the Geomasking.

Table 5.6: Averagely moved distances Geomasking

n_{closer}	d
4,000	2,009.598 m
16,000	3,788.147 m
64,000	6,439.553 m
256,000	8,668.326 m

6 Results

Since the Geomasking has a fundamentally different approach in comparison to the other synthesis methods, special conditions apply for this algorithm and the results of the utility and risk evaluation metrics cannot always be compared to the results of the other synthesizers. For simplification, in the following chapter we will refer to the CART, the Random Forest, and the Copula Synthesizers as full synthesizers, as they synthesize the entire data set (except for the geospatial attribute).

6.1 Results of Risk Evaluation

6.1.1 Results of Privacy Attack

In figure 6.1 we present the accuracy and balanced accuracy, accumulated with equal weights over every stratum and therefore independent from the number of individuals per stratum. We choose this way of accumulation in order for the strata with a lower number of individuals to have a stronger impact on the evaluation metric, since smaller strata tend to be more in danger of a higher re-identification risk, as explained in chapter 3.

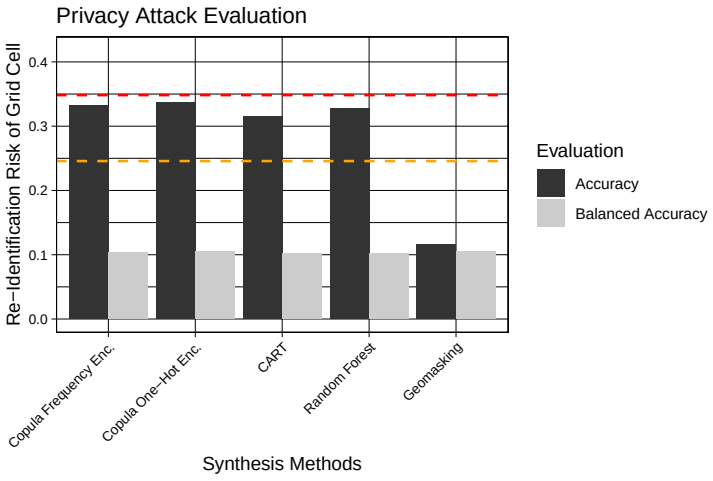


Figure 6.1: Accuracy and Balanced Accuracy of Privacy Attack

The figure (6.1) illustrates that for all full synthesizers, the accuracy is far higher than the balanced accuracy. This means that the Privacy Attack tends to classify bigger classes more often correctly than smaller classes, which can be explained by Random Forests being biased towards bigger classes if trained on data sets of different class sizes. Comparing the full synthesizers with each other, we do not see any major differences. The accuracy for the CART Synthesizer is only slightly lower than for the other three, while all full synthesizers have roughly the same balanced accuracy. The Geomasking also has a balanced accuracy equal to the full synthesizers. However, the accuracy is almost as low as the balanced accuracy. This is probably the case because the Geomasking moves the individuals around the different grid cells within the entire stratum, which results in them being almost equally distributed over the area of the stratum. For interpreting the absolute values of the accuracy, we added two

horizontal dashed lines to the figure. The orange line represents the accuracy of a Privacy Attack algorithm that predicts the grid cell by sampling from a multinomial distribution with weights of the actual class sizes. The red line represents the accuracy of a Privacy Attack algorithm that always predicts the biggest class. Therefore, both lines stand for a certain way of guessing and should point out that using Privacy Attack the absolute re-identification risk of the examined synthesis methods is not or not much higher than if the spatial identifier e.g. the grid cell was predicted by guessing.

6.1.2 Results of Population Uniqueness

We will now investigate the results of the Population Uniqueness analysis, based on figure 6.2. On the horizontal axis we can see which attributes are added in which order to the subset of interest. To give an example: The values for HH_GROESSE_PERSON represent the results of an analysis for a subset of HH_GROESSE_PERSON, RAUMANZAHL_KLASS, and ALTER_01JS. On the vertical axis we can see the proportion of uniquely re-identifiable individuals for the given subset, grouped by synthesis method. The lighter colors (Mean) represent the average proportion of all strata, while the darker colors (Max) depict the highest proportion of re-identified individuals out of all strata.

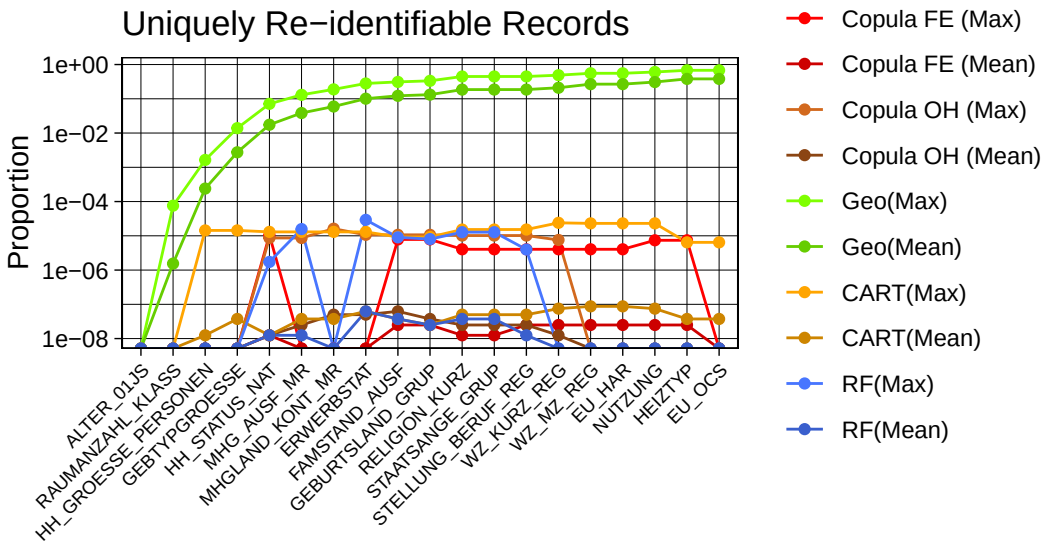


Figure 6.2: Proportion of Uniquely Re-identifiable Records

As shown in the figure (6.2), the re-identification risk based on Population Uniqueness is negligible for all full synthesizers. On the one hand, the chances for a unique record increase with an increasing number of attributes in the observed subset. On the other hand, with an increasing number of attributes there is a drastically decreasing chance of the original and synthetic records being identical. This makes it very unlikely for those four synthesizers to generate uniquely re-identifiable records. Because the Geomasking only alters the spatial attribute, the other attributes in the synthetic data set are always identical to the original attributes, and the more attributes are part of the investigated subset, the more this proportion increases. As a consequence, there are close to 100% uniquely re-identifiable records in cases where the attacker already knows almost all of the non-sensitive attribute

values. However, we need to add that this risk only refers to the non-sensitive records in the Geomasked data set, since for the Geomasking only one data set is to be published. After re-identifying an individual, the attacker will still be confronted with the shifted spatial identifiers.

6.2 Results of Utility Evaluation

The following metrics were, except for the use case, calculated using the [NRD16] package in R. In order to get a first impression of the utility of the different synthesis methods, we will take a look at the results of the pMSE.

Table 6.1: pMSE Results

Synthesis	Copula Frequency Enc.	Copula One-Hot Enc.	CART	Random Forests	Geomasking
pMSE	0.24999	0.24226	0.19816	0.21867	0.10040

We can see that the pMSE of the Copula Synthesis with Frequency Encoding is almost as high as 0.25, which is the highest value possible. This could indicate a very poor similarity of the overall distribution between D_{geo} and D_{orig}. The value of the Copula Synthesizer with One-Hot Encoding of 0.242 is also very high, but already notably further away from 0.25. For the Random Forest Synthesizer, we get an even lower value of 0.219 and for the CART Synthesizer we obtain the lowest value for all full synthesizers, which is 0.198. As explained in section 4.2, the pMSE is highly dependent on the discriminator model and naturally increases with the number of predictors, which makes it impossible to make statements about the absolute utility only based on the pMSE. In total, the Geomasking algorithm delivers by far the lowest value, which is presumably due to the fact that only the spatial identifier was altered.

6.2.1 Preserving Univariate Distributions

To learn more about the synthetic distributions, we will now take a look at figure 6.3 of the univariate distributions, based on the VW-Test.

The test statistic was calculated for every stratum separately. The results in the plots are ordered by the number of individuals in the respective stratum, with low numbers on the bottom. We can see that the tendencies found for the pMSE continue. For almost all variables, the test statistic is visibly higher for D_{copFE} than for D_{copOH}, while D_{rf} and D_{cart} have even lower values. Moreover, among the attributes where the values are comparably low anyway, D_{cart} maintains lower values than D_{rf}. We also can see that the same attributes for all four sets have higher values than the others. The fact that high numbers of individuals per stratum lead to a higher value of the test statistic is influenced by the higher number of counts for the different classes, which leads to higher values of the test statistics. It does not mean that the utility is better or higher for less populated strata. The variables in the plots are ordered by their number of unique values, increasing from left to right. In accordance with our expectations, the test statistic values for variables with a higher number of unique

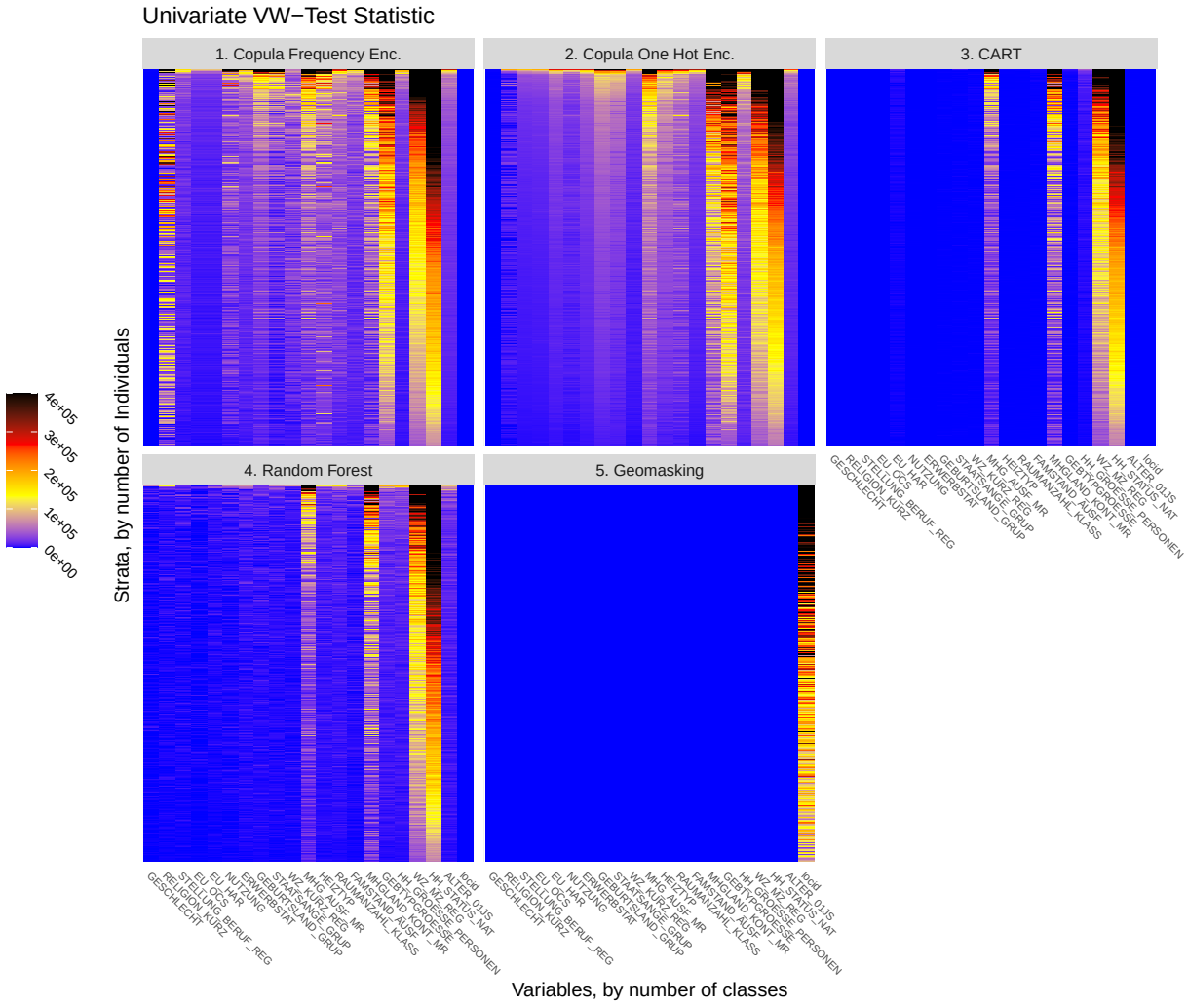


Figure 6.3: Univariate WV-Test Statistic

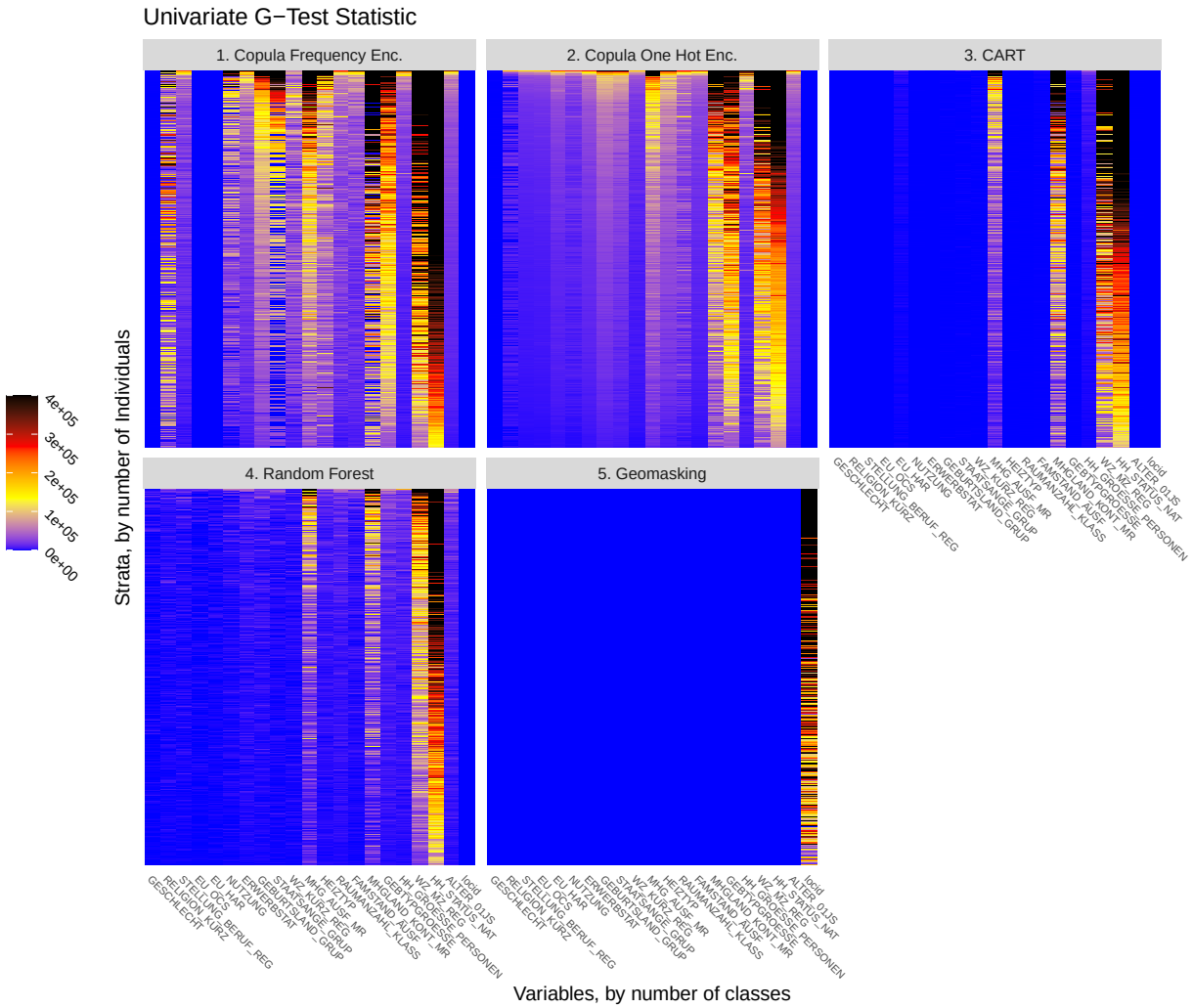


Figure 6.4: Univariate G-Test Statistic

values are higher. However, among variables with similar numbers of unique values, some appear to be affected more than others.

Since categorical attributes have to undergo an encoding and decoding process, one thing that we might expect to find is that for categorical attributes the distributions might generally not be preserved as well as for integer attributes. Moreover, this is one of the reasons we chose a CART and a Random Forest Synthesizer as alternative methods in the first place. Details on that can be found in section 2.2.4. For all full synthesizers, we can see that for instance ALTER_01JS and HH_GROESSE_PERSONEN both show way lower values than their neighboring attributes, while both are integer attributes. On the other hand, the three attributes with the most black patches in the plots, HH_STATUS_NAT, WZ_MZ_REG, and MHGLAND_AUSF_MR, are categorical variables. Generally, all integer attributes have comparably low values, except for GEBTYPGROESSE. Furthermore, many categorical attributes have very low values, such as FAMSTAND_AUSF.

For gathering further information on whether the data type of a variable could influence how well the distribution is maintained, we will also take a look at the results of the G-Test

for the univariate distributions in figure 6.4. Here we can see that the general tendencies of the test statistic values are the same as for the VW-Test. By comparing the variables, we again find how GEBTYPGROESSE as the only integer variable has comparably high values. Moreover, FAMSTAND_AUSF again shows comparably low values. In order to learn more about the influence of the data type, we will now take a more detailed look at the figure of the Random Forest Synthesizer. If our theory about the influence of the different data types was true, we would expect to find the categorical attributes to improve more than the integer attributes between the Copula Method with Frequency Encoding and the Random Forest Synthesizer. However, the figure reveals that such a pattern is not obvious. There are strong reductions in the VW-Test statistic for categorical attributes, like RELIGION_KURZ or WZ_MZ_REG, but equally also for the integer variable GEBTYPGROESSE.

In figure 2.6, we could observe how GEBTYPGROESSE is far away from following a normal distribution, which is why we would expect the test statistic of this value to improve largely by modeling it with the CART or Random Forest Method. In figure 6.3 we can confirm that the test statistic of this variable became considerably lower.

The results of the Geomasking synthesizer are as expected: For both test statistics, the values are extremely high for the spatial identifier and zero for all other variables. Similarly, the values for the spatial identifiers for the full synthesizers are zero, since no alteration was made.

A further interesting finding is that for the G-Test, EU_OCS has extremely low values close to zero for the Copula Synthesizers with Frequency Encoding, compared to the VW-Test. To investigate this phenomenon, we will compare the value counts before and after synthesis.

Table 6.2: EU_OCS Value Counts

Class	OCS_01	OCS_02	99
True	78,648,050	1,276	1,560,671
Copula Frequency Enc.	80,209,997	-	-
Copula One-Hot Enc.	74,792,789	60,038	5,357,170
Random Forest	78,612,001	1,768	1,596,228

Table 6.2 reveals that the synthetic attribute of the Copula Method with Frequency Encoding only contains class OCS_01. The synthesis did not preserve the distribution well. However, as the G-Test only considers classes where the synthetic counts are greater than zero, the resulting test statistic values do not reflect the absence of the classes OCS_02 and 99. Therefore, they also do not reflect the bad preservation of the distribution of the attribute EU_OCS in the synthetic data.

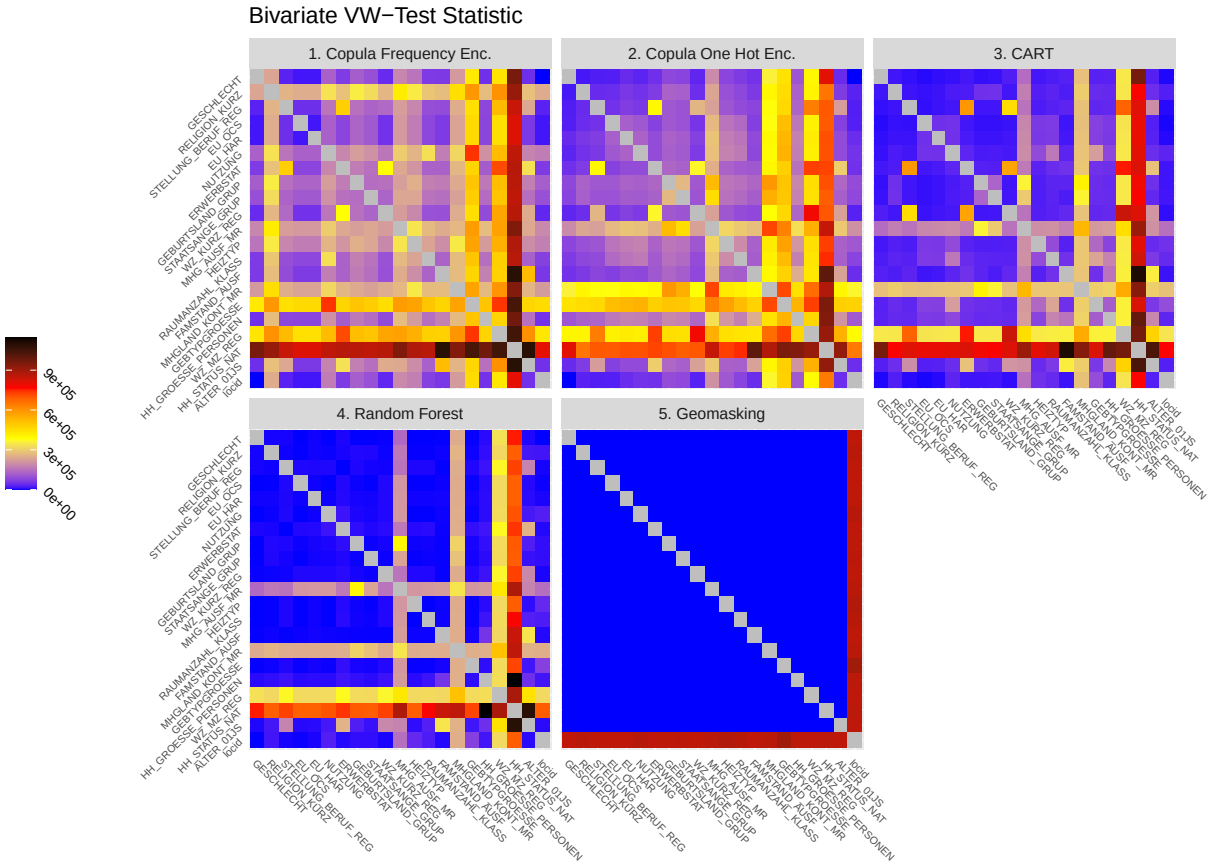


Figure 6.5: Bivariate VW-Test Statistic

6.2.2 Preserving Multivariate Distributions

In the following, we want to figure out how well the multivariate distributions are preserved for the different synthesis methods and what kind of distributions might be troubling for the synthesizers. For this purpose, we will now look at the results of the VW-Test in figure 6.5 for bivariate distributions. The results of the bivariate G-Test can be found in the appendix in figure 7.1. For both test statistics, the figures only show minor differences.

Generally, this figure (6.5) indicates the same tendencies as the univariate distribution plots. The same variables tend to have higher or lower test statistic values than others. However, there seem to be some variable combinations where the distribution is preserved worse than for the surrounding combinations. In section 2.2.2, we took a closer look at the bivariate distribution between HH_STATUS_NAT and FAMSTAND_AUSF and discussed why the Copula Method with Frequency Encoding would be expected to struggle modeling this distribution. We can see from the figure that this combination indeed does have a comparably high test statistic value. However, this combination of attributes also shows comparably high test statistic values for the other synthesizers. This means that the other synthesizers also do not preserve this distribution properly. Therefore, distributions where the classes have strong and complex dependencies to each other might still be challenging for the tree-based synthesizers, even though we expected these kind of distributions to be the key strength of tree-based synthesizers.

After all, we want to keep in mind that, regarding the bivariate plots, mainly the distributions including the geospatial attribute (locid) are of special interest, since a potential data product should allow precise geospatial analysis. We can see how these distributions are by far the worst for the Geomasking. This could be a hint that the full synthesizers could actually provide an advantage for low-level spatial analysis, compared to Geomasking.

6.2.3 Results Use Case: Unemployment Rate

The results of the use case, as defined in section 4.3, show us how useful the results of a low-level spatial analysis on the different synthetic data sets would be. First, we examine the aggregated results in table 6.3, where the results for each grid cell (unique value of spatial identifier GITTER_ID_10KM) were weighted by the number of individuals per grid cell. By far the highest relative error is to be found for the Copula Method with Frequency Encoding. On average, the calculated unemployment rate has a relative error of 125% compared to the true value. The results for the other four methods are notably better, while the Copula Method with One-Hot Encoding still performs worse than the tree-based methods and the Geomasking. The best results are obtained for the CART Synthesizer, with only 19.0% relative error. For the CI Overlap, we observe how the value of the Copula Method with One-Hot Encoding is extremely low (0.4%). The best CI Overlap (13.5%) is again provided by the CART Synthesizer.

Table 6.3: Use Case Results Aggregated

Synthesis	Copula Frequency Enc.	Copula One-Hot Enc.	CART	Random Forests	Geomasking
Relative Error	1.2575	0.4394	0.1903	0.3696	0.3580
CI Overlap	0.0651	0.0040	0.1351	0.0831	0.1158

Figure 6.6 shows relative error and CI Overlap disaggregated by grid cell. These cells are ordered by number of individuals on the horizontal axis.

First, we notice that there are many grid cells where all full synthesizers have no values at all. This happens because these grid cells are actually uninhabited, but since the Geomasking moves the locations around within the entire stratum, it often happens that some of them get moved into uninhabited grid cells. For the same reasons, some grid cells show some gray lines for the Geomasking, while in the same grid cell the other synthesizers have counts: The Geomasking moved all individuals away. The relative errors in figure 6.6 for the different synthesizers correspond with the aggregated results in table 6.3. Moreover, for the relative error, there does not seem to be an obvious tendency regarding the number of individuals per grid cell. Regarding the CI overlap, we find that grid cells with a very low number of individuals tend to have high values, which go up to 100%. This is often the case when in

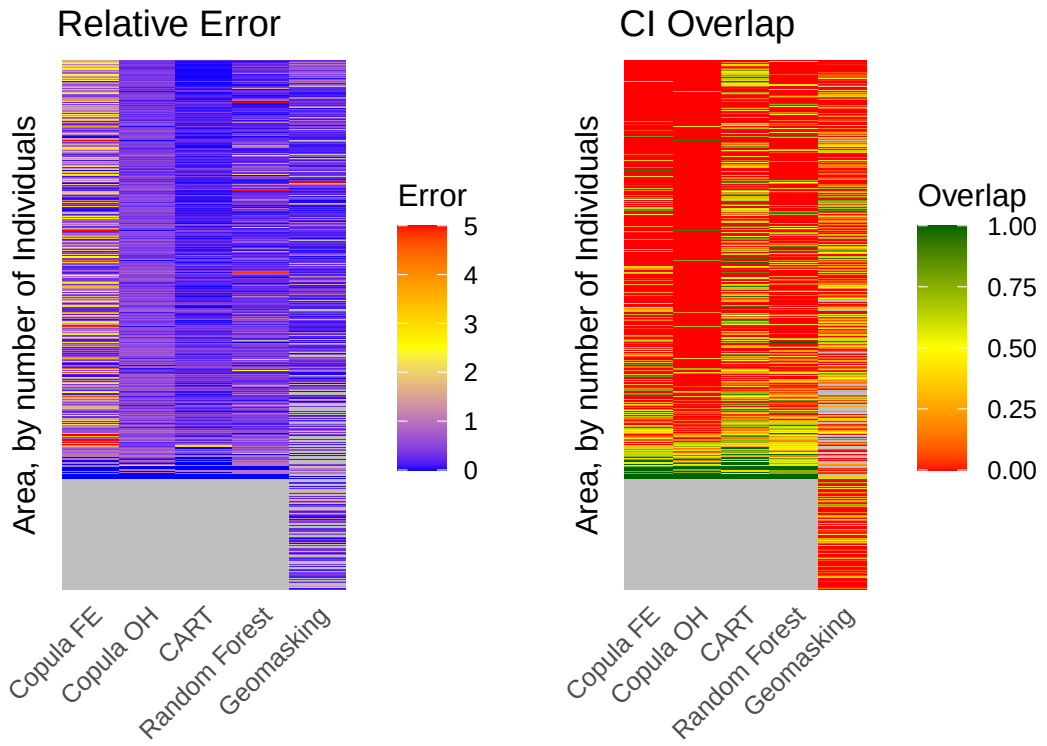


Figure 6.6: Use Case Results by Grid Cell

those cells there are only a few individuals living, which are all in employment, as correctly reflected in the synthetic data. Therefore, for the true and synthetic binomial distributions of unemployed individuals $p = 0$ applies, which makes the 95% CI $[0, 0]$ and results in a 100% overlap. Speaking of absolute figures, however, the CI Overlaps are rather small for all synthesizers. Especially for more populated grid cells, the overlap is often equal or close to zero, which means there is no overlap at all. For the Copula Method with One-Hot Encoding, the CI Overlaps are even worse compared to the other synthesizers.

In order to investigate the CI Overlap further, we will consider figure 6.7, which compares the synthetic and true unemployment rates together with their 95% confidence interval. As an example, we use the grid cells of only one randomly selected stratum for this comparison.

On the horizontal axis we can see the grid cells ordered by number of inhabitants, in ascending order from left to right, while on the vertical axis we see the unemployment rate. Since we model the employment status of an individual with a binomial distribution, as explained in section 4.3, the CIs are shrinking for more populous grid cells. Therefore, even though the relative error from the synthetic data might not be that severe, the CIs are so small that they do not overlap.

This figure (6.7) also gives us more details about the performance of the different synthesizers, especially, which of them produce a bias for the estimators of the unemployment rate. Obviously, the strongest bias was generated by the Copula Synthesis with One-Hot Encoding, which also has the overall worst CI Overlap. The proportions of unemployed

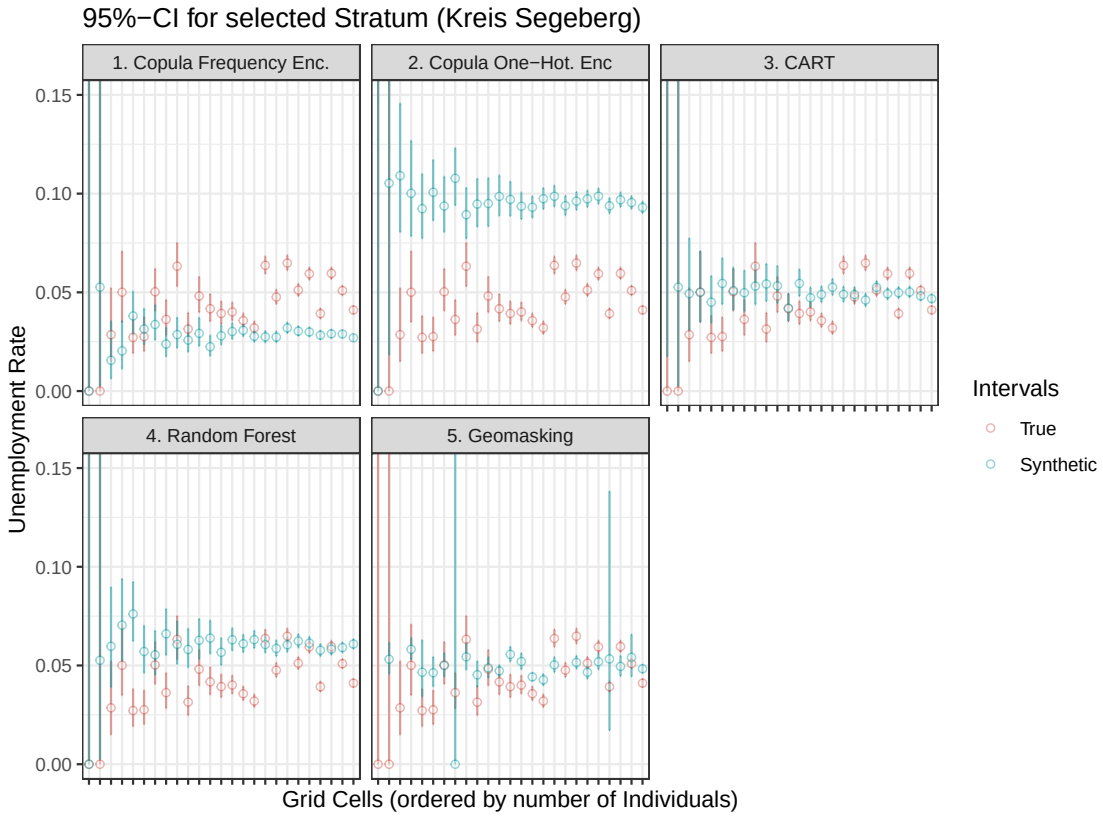


Figure 6.7: 95% CI: Comparison between True and Synthetic Data

individuals in D_{copOH} are so high that not even the less populated grid cells produce any overlap. This finding matches with the results from section 2.2.2, where the method for reversing the One-Hot Encoding was discussed. From figure 2.5 we could learn that our reversing method is biased towards small classes, thus, creating the proportion of small classes in the synthetic data too high. The bad results for the CI Overlap together with the plotted confidence intervals could therefore point out that the weakness of the reversing process of the One-Hot Encoding has a huge negative influence on the synthetic data. Also, the Copula Method with Frequency Encoding and the Random Forest Synthesizer generate a visible bias. The only two synthesizers where the estimators of the unemployment rate do not have a visible bias are the Geomasking and the CART Synthesizer. Moreover, the estimators of the Geomasking seem to have a bigger variance than those of the CART Synthesizers.

Another thing we visually notice is that for none of the synthesizers, the unemployment rates on the sub-stratum level seem to be preserved. This would mean that in this project the conditioning on the spatial identifier did not influence the synthetic data noticeably.

To sum up, the CART Synthesizer delivers the best results for this use case, which are notably better than the results of the Geomasking. The worst results are generated by the two Copula methods. The Copula Method with One-Hot Encoding seems to suffer severely from the bias caused by the reversing of the One-Hot Encoding, while the Copula Synthesis with Frequency Encoding produces huge relative errors.

7 Discussion

In this project, we have evaluated five different synthesis methods, which are two Copula Synthesizers with Frequency Encoding and One-Hot Encoding, respectively, as well as a Random Forest Synthesizer, a CART Synthesizer, and Geomasking. The purpose was to analyze the potential that synthetic versions of official micro data with exact geospatial attributes offer for the German official statistics and science.

Generally, the evaluated risk metrics suggest a negligible re-identification risk for all synthesizers, except for the Geomasking. Therefore, the final assessment of which model has the highest potential for the German official statistics and science is mainly based on the utility evaluation. Out of all synthesizers from this project, the CART Synthesizer produced the most promising results based on global and specific utility metrics and the evaluated use case. The Random Forest Synthesizer performed notably worse than the CART Synthesizer but still better than the Copula Synthesizers. For the Copula Synthesizer with Frequency encoding, we could see from an example that the proportion of small classes in the synthetic data is constantly too low, which means that the respective estimator in the synthesis model is biased. The estimators of the proportion of small classes from the Copula Method with One-Hot Encoding are affected by an even larger bias than the Random Forest Synthesizer, while the Copula Synthesizer with Frequency Encoding performed the worst in general. The Geomasking comes with an extremely uncontrollable re-identification risk, so even though this synthesizer produces overall competitive utility metrics, it cannot be considered an alternative to the CART Synthesizer.

During the project, we were able to determine the following reasons, why the Copula Method with Frequency Encoding did perform worse than the other models. Firstly, the Frequency Encoding produces uniformly distributed marginal distributions of the categorical attributes and expects the synthetic samples to be uniformly distributed. However, the synthesis algorithm models all attributes with a normal distribution. Secondly, the integer attributes also often do not follow a normal distribution, which also results in poorly preserved marginal distributions in the synthetic data. Finally, the complexity of the relationships between the categorical attributes is too high to be properly modeled (after Frequency Encoding) by just the covariance, which is the only parameter in the Copula Model that is capable of describing the relationships between different attributes. Worth mentioning is that for calculating this synthesis model we had to make some distributional assumptions regarding the covariances in order to deal with the missing values in the original data. The required assumption was that the covariance of two attributes is equal to the covariance of only the missing values among those two attributes. We cannot definitively determine from our results whether this assumption is justified or whether this phenomenon considerably impacts the modeling properties of the synthesis. To sum up, the parametric assumptions of the Copula Model seem to be too restrictive to allow the synthetic data to fit the distribution of the original data sufficiently.

[T K23] presents the Copula Synthesis with Frequency Encoding as promising method for providing synthetic geo-referenced micro data. According to the authors' findings, this synthesis method provides a lower re-identification risk and at the same time a higher utility than Geomasking. While our project also found a lower re-identification risk for the Copula Synthesis with Frequency Encoding compared to the Geomasking, the results of our project regarding the utility do not support their findings. One possible reason for these contradictory results is that the relative number of numerical attributes in our data was relatively low, which means that most attributes had to be Frequency Encoded, which due to the modeling properties of this encoding method might have had a negative influence on preserving the distribution of the data. By their application of the algorithm, only roughly 30% of individuals got assigned a new spatial identifier (zip code), which leaves the entire records of 70% of individuals completely unmodified. This might also be a further explanation for [T K23] finding a higher re-identification risk for the Geomasking. For comparison, in our project more than 80% of individuals were assigned a new spatial identifier during the Geomasking, represented by the attribute GITTER_ID_10km. Another likely reason for the different findings is that we did not perform any sampling at all and therefore performed the data synthesis always on the entire data set. In contrast, [T K23] used a two-stage stratified sampling technique, which might naturally have an effect on the risk and utility metrics. Especially the notably higher risk of re-identification via unique records found in [T K23] has most likely been caused by fewer individuals being present in the data source there.

The authors of [T K23] also examined the Copula Synthesis for One-Hot Encoding, with the result that Frequency Encoding leads to a slightly better utility and to better computation costs, while at the same time avoiding the issue of potential multicollinearities in the encoded data. However, in our project we found that One-Hot Encoding leads to a notably higher utility than Frequency Encoding. One possible reason for the different results is that the data source of [T K23] includes a notably higher number of attributes (106, instead of 26), which might have caused in regard to One-Hot Encoding even bigger multicollinearity issues and computational costs. We also want to add that the way [T K23] handles potential multicollinearities, as well as the way they reversed the One-Hot Encoding after drawing the synthetic samples, is not derivable from their paper. In our project, we therefore might have used completely different approaches, markedly lowering the comparability of their and our results. We noticed that our approach of reversing the One-Hot Encoding caused biased estimators of the proportions of small classes in the synthetic data. More precisely, the relative size of small classes in the synthetic data is too high. The method for reversing the One-Hot Encoding as used in our project is probably the main weakness of the Copula Synthesis with One-Hot Encoding. Further weaknesses may include the modeling of the One-Hot Encoded attributes with a normal distribution, since for One-Hot Encoded attributes a normal distribution is most likely not given. Moreover, a weakness could be the slightly reduced covariances caused by the treatment of the multicollinearities.

According to expectations, the CART and Random Forest Synthesizers, which have the capability of modeling complex categorical multivariate relationships without underlying parametric assumptions, are much better capable of transporting the original distribution to the synthetic data than the Copula Methods, which results in them producing notably better results for the utility metrics. Furthermore, our results support ideas from [DR11] and [DH23] that in data synthesis, CART Synthesizers usually outperform other synthesis models.

Moreover, unlike in [DR11], we could not detect a higher re-identification risk of our CART Synthesizer in comparison to our Random Forest Synthesizer.

One of the aspects that according to our observations did not show a significant influence was the data type of the attributes. The categorical attributes did not show notably higher values for the test statistics than the integer attributes, which applied to all synthesizers, regardless of whether the synthesizer used encoding. Hyperparameters also do not seem to impact the performance of the tree-based models considerably. Especially for the CART Synthesizer, this influence was found to be negligible for the values over which the depth of the trees got optimized. Only for the Geomasking, a certain influence of the hyperparameters was noticeable.

There are other synthesis models without parametric assumptions that we could have possibly considered and want to present briefly at this point. According to [DH23], literature about GANs *(Generative Adversarial Networks)*, first introduced by [Goo+14], is growing rapidly in regards to data synthesis. GANs consist of two networks competing with each other. Based on white noise, the so-called generator tries to generate an artificial object as close to the training data as possible. At the same time, the so-called discriminator tries to discriminate between the original and the synthetic object, while minimizing its own classification error. Since a low classification error of the discriminator leads to a high penalty for the generator, the network learns to generate synthetic data that is indistinguishable from real data. That means, if we train GANs on a data set, the generator can produce synthetic data samples from it. Another type of generative models for data synthesis would be VAEs *(Variational Auto Encoders)*, introduced by [KW22] and first published in 2013. These models train an encoder-decoder architecture projecting into and from a so-called latent space. During the encoding process, noise is added to the training observations. This has the effect that one training record is not only represented by one exact position in the latent space, but by an entire distribution (e.g. the area close by). Moreover, the latent space is locally restricted by being forced to follow a certain distribution (e.g. the multidimensional standard normal distribution). For the predictions, this means that independently from where in the latent space we decode a sample, we will always receive a meaningful output. While this property of VAEs can be used for generating synthetic objects like artificial images, it can also be used to generate synthetic data records. Just as for other generative models, the big disadvantages of the two presented ones are the huge amount of required training data and high training costs. Another reason why they are not suitable for our purpose is that in their base form they do not allow the generation of synthetic samples based on predefined conditions via conditional sampling. Therefore, they cannot be used for the synthesis of partially synthetic data. Finally, since they are based on neural networks they require numerical data, which would require an additional encoding and decoding process for our predominantly categorical data set. Considering the presented disadvantages, we decided against evaluating generative models like GANs and VAEs in this project.

Since even for the CART Synthesizer the re-identification risk is negligible in absolute terms, there might still be potential ways of obtaining synthesis models that produce synthetic data with higher utility and yet with still an acceptable re-identification risk. One conceivable option may be to use a synthesis model that is able to model the distribution of the original data more precisely than the CART Synthesizer. This could be a model based on Ensemble

Methods, like a Random Forests Synthesizer, but with predictions from all trees instead of only OOB predictions. Moreover, very interesting to observe was that even for the CART Synthesizer, the conditioning on the spatial identifier did not show any noticeable influence on the synthetic data. This would mean that for producing more accurate low-level synthetic data the spatial identifier of the grid cells might require more influence in the synthesis process. Another aspect that makes especially controlling of the re-identification risk more challenging is the varying sizes of the strata and varying number of individuals per 10x10km grid cell. If the spatial identifier of an individual living in a 10x10km grid cell with only one inhabited building is re-identified correctly, the address of this individual is re-identified as well. In order to avoid this kind of problem, it would be desirable to work with strata that all have the same population and the same number of equal-sized sub-strata disaggregations. Since the German administrative disaggregations do not provide such criteria, an alternative would be to use an algorithm that disaggregates Germany into smaller areas of equal population, which could be used as strata. If these strata were disaggregated further into a fixed number of smaller same-sized sub-strata areas, these could be used as spatial identifiers (in this project: GITTER_ID_10KM).

Finally, we want to give a statement on the main topic of this project, which is an assessment of the potential that data synthesis offers for providing sensitive official micro data with exact geo-referenced attributes. First of all, regarding the generation of two data files as proposed by [T K23], it seems unlikely that access to both of the files could be provided at the same time, due to increased data privacy risks. However, providing controlled access by technically ensuring that they can only access one of the data sets at a time would be conceivable.

For providing a controlled data access for independent scientific research institutions, the Research Data Center of the German statistical office offers different ways of access, depending on the level of anonymity of the requested data source, as described in [Rot19] and [Züh+05]. Onsite access allows analyzing microdata with the highest analytic potential because only direct identifiers are removed. Researchers either need to visit the so-called Safe Center, a workplace within the Statistical Offices, or use Remote Execution. For Remote Execution, researchers program their codes based on publicly available absolutely anonymous files that only give information on the structure of the data. Hence, they do not produce interpretable results before the code is run by employees of the Research Data Center on the true data. Moreover, standardized and de facto anonymous data sets can be downloaded as SUF (Scientific Use File), after a registration process. Such data files contain samples of the original data sources, where potentially sensitive attributes, such as spatial identifiers are either anonymized, for instance via aggregation, or completely removed.

A synthetic data set, as discussed in this project, could potentially be considered de facto anonymous, which means that considering its level of anonymity it would have to be positioned between a SUF and onsite data. According to the given regulations on data privacy of German official microdata, this could allow for such a synthetic data set to be accessible as RSUF (Remote Scientific Use File) via Remote Access. For Remote Access, there are several technical, organizational, and contractual measures to ensure data privacy (e.g., access is only possible for a specified range of IP addresses from the network of the research institute, up- and downloading of data is prohibited and all performed runs on the

data are automatically logged, de-anonymization is contractually forbidden and leads to contractual penalties). Instead of requiring the researchers to commute to the office buildings of the German Statistical Office, synthetic data accessible as RSUF would allow researchers to work directly from their research institutes, which would be advantageous as long as the synthetic data provides them with an adequate alternative to the original data source. A provision of the described data source as SUF seems to be rather unlikely because of the strict privacy-ensuring regulations for German official microdata.

Appendix

Number of PSUs per stratum on stratified sample from paper [T K23], compared to on full data:

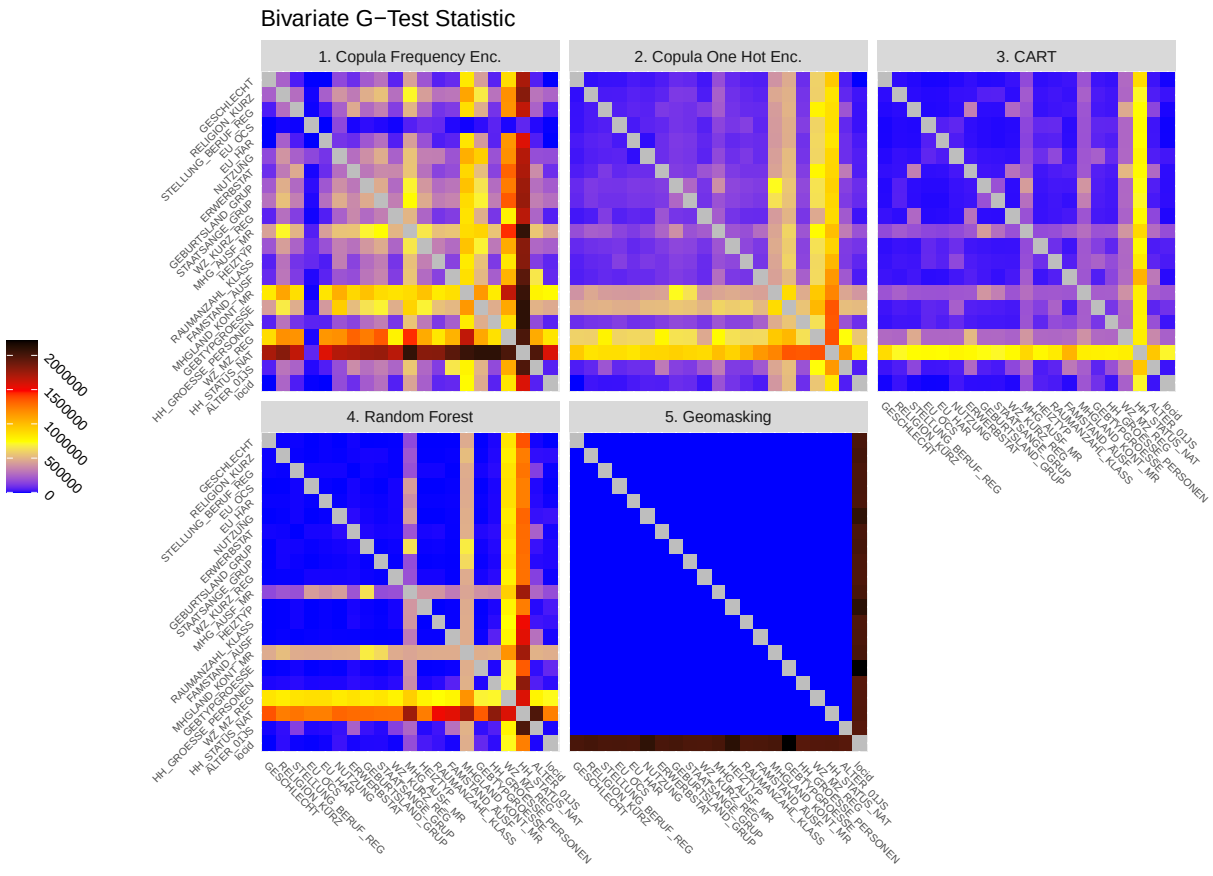
$$\text{On Sample} : \frac{n_{PSU}^{sample}}{n_{strata}} = \frac{123}{12} \approx 10 \tag{7.0.1}$$

$$\text{On Full Data} : \frac{n_{PSU}^{true}}{n_{strata}} = \frac{767}{12} \approx 64 \tag{7.0.2}$$

Individuals per PSU on 2% stratified sample from paper [T K23], compared to on full data:

$$\text{On Sample} : N \cdot p_{sample} \cdot \frac{1}{n_{PSU}} = 427.830 \cdot 0.02 \cdot \frac{1}{123} \approx 70 \tag{7.0.3}$$

$$\text{On Full Data} : \frac{N}{n_{PSU}} = \frac{427.830}{123} \approx 558 \tag{7.0.4}$$



h

Figure 7.1: Bivariate G-Test Statistic

Table 7.1: Data Source Overview

Variable	Meaning	Data Type	Unique Values	Missing Values	Missing Values (Raw Data)
PERSON_ID	Individual ID	integer	80,209,997	0	0
ALTER_01JS	age	integer	101	0	0
GESCHLECHT	sex	integer	2	0	0
RAUMANZAHL_KLASS	number of rooms	integer	7	1,537,015	1,546,713
HH_GROESSE_PERSON	household size	integer	11	1,537,015	1,546,713
GEBTYPGROESSE	Size of home building	integer	10	1,537,015	1,546,713
AGS_12	ID for commune	factor	11,491	0	0
HH_STATUS_NAT	house hold status	factor	16	0	0
MHG_AUSF_MR	migration country	factor	6	0	0
MHGLAND_KONT_MR	continent of origin	factor	9	0	0
ERWERBSTAT	employment status	factor	5	0	0
FAMSTAND_AUSF	family status	factor	8	0	0
GEBURTSLAND_GRUP	country of birth	factor	5	0	0
RELIGION_KURZ	religion	factor	3	0	0
STAATSANGE_GRUP	citizenship	factor	5	0	0
STELLUNG_BERUF_REG	employment position	factor	3	0	0
WZ_KURZ_REG	Employment Sector	factor	5	0	0
WZ_MZ_REG	Empl. Sub-Sector	factor	12	0	0
EU_HAR	Type of accommodation	factor	4	0	0
NUTZUNG	—	factor	4	0	1,546,713
HEIZTYP	Way of heating	factor	7	0	1,546,713
EU_OCS	—	factor	3	0	1,546,713
GITTER_ID_100M	ID of 100x100m square	factor	3,296,697	0	9,698
GITTER_ID_1KM	ID of 1x1km square	factor	217,992	0	9,698
GITTER_ID_10KM	ID of 10x10km square	factor	3,826	0	9,698
REGION_KREIS	ID of county	factor	412	0	9,698

Table 7.2: Constraints for Conditional Sampling

ALTER_01JS	$x \geq -0.5$
GESCHLECHT	$x \in [0.5, 2.5]$
RAUMANZAHL_KLASS	$x \in [0.5, 7.5]$
HH_GROESSE_PERSON	$x \in [0.5, 11.5]$
GEBTYPGROESSE	$x \in [0.5, 10.5]$
All Categorical Attributes	$x \in [0, 1]$

All non-categorical attributes are after sampling rounded to the next integer.

List of Figures

List of Tables

Bibliography

[AHJ13] Vincent Audigier, François Husson und Julie Josse. *A principal components method to impute missing values for mixed data*. 2013. arXiv: 1301.4797 [stat.AP]. URL: https://arxiv.org/abs/1301.4797.

[CR10] Gregory Caiola und Jerome P. Reiter. *Random Forests for Generating Partially Synthetic, Categorical Data*. 2010. URL: https://api.semanticscholar.org/CorpusID:7953157.

[DH20] Jörg Drechsler und Jingchen Hu. *Synthesizing Geocodes to Facilitate Access to Detailed Geographical Information in Large-Scale Administrative Data*. Dez. 2020. DOI: 10.1093/jssam/smaa035. eprint: https://academic.oup.com/jssam/article-pdf/9/3/523/39308315/smaa035_supplementary_data.pdf. URL: https://doi.org/10.1093/jssam/smaa035.

[DH23] Joerg Drechsler und Anna-Carolina Haensch. *30 Years of Synthetic Data*. 2023. arXiv: 2304.02107 [cs.CR]. URL: https://arxiv.org/abs/2304.02107.

[DR11] Jörg Drechsler und Jerome Reiter. *An empirical evaluation of easily implemented, nonparametric methods for generating synthetic datasets*. Dez. 2011. DOI: 10.1016/j.csda.2011.06.006.

[Goo+14] Ian J. Goodfellow u. a. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML]. URL: https://arxiv.org/abs/1406.2661.

[Ish+08] H. Ishwaran u. a. *Random survival forests*. 2008. URL: https://arXiv.org/abs/0811.1645v1.

[KW22] Diederik P Kingma und Max Welling. *Auto-Encoding Variational Bayes*. 2022. arXiv: 1312.6114 [stat.ML]. URL: https://arxiv.org/abs/1312.6114.

[Lit93] R. J. A. Little. *Statistical analysis of masked data*. 1993. URL: https://www.proquest.com/scholarly-journals/statistical-analysis-masked-data/docview/1266808565/se-2.

[Man77] Maudsley AA. Mansfield P. *Medical imaging by NMR.* 1977. DOI: 10.1259/0007-1285-50-591-188. URL: http://dx.doi.org/10.1561/0400000042.

[MK09] A. Oganian M. J. Woo J. P. Reiter und A. F. Karr. *Global measures of data utility for microdata masked for disclosure limitation*. 2009.

[NRD16] Beata Nowok, Gillian M. Raab und Chris Dibben. *synthpop: Bespoke Creation of Synthetic Data in R*. 2016. DOI: 10.18637/jss.v074.i11.

[Rei05] J. Reiter. *Using CART to Generate Partially Synthetic, Public Use Microdata*. Jan. 2005.

[RM09] Jerome P. Reiter und Robin Mitra. *Estimating Risks of Identification Disclosure in Partially Synthetic Data*. 2009. DOI: 10.29012/jpc.v1i1.567. URL: https://journalprivacyconfidentiality.org/index.php/jpc/article/view/567.

[RND21] Gillian M Raab, Beata Nowok und Chris Dibben. *Assessing, visualizing and improving the utility of synthetic data*. 2021. arXiv: 2109.12717 [stat.CO]. URL: https://arxiv.org/abs/2109.12717.

[RR83] PAUL R. ROSENBAUM und DONALD B. RUBIN. *The central role of the propensity score in observational studies for causal effects*. Apr. 1983. DOI: 10.1093/biomet/70.1.41. eprint: https://academic.oup.com/biomet/article-pdf/70/1/41/662954/70-1-41.pdf. URL: https://doi.org/10.1093/biomet/70.1.41.

[Rub81] Donald B. Rubin. *The Bayesian Bootstrap*. 1981. DOI: 10.1214/aos/1176345338. URL: https://doi.org/10.1214/aos/1176345338.

[Rub93] D.B. Rubin. *Statistical Disclosure Limitation*. 1993.

[Skl59] Abe Sklar. *Fonctions de répartition à n dimensions et leurs marges*. 1959.

[Sno+18] J. Snoke u. a. *General and specific utility measures for synthetic data*. 2018.

[T K23] T. Schmid T. Koebe A. Arias-Salazar. *Releasing survey microdata with exact cluster locations and additional privacy safeguards*. 2023. URL: https://doi.org/10.1057/s41599-023-01694-y.

[VW01] David Voas und Paul Williamson. *Evaluating Goodness-of-Fit Measures for Synthetic Microdata*. 2001. URL: https://api.semanticscholar.org/CorpusID:122381913.

Data and Metadata

[Bay16] Forschungsdatenzentrum des Bayerischen Landesamts für Statistik. *Schlüsselverzeichnis für das Zensus 2011 – Produkt 3 – Gesamtdatensatz zur Nutzung über die On-Site Zugangswege*. Mai 2016. URL: https://www.forschungsdatenzentrum.de/sites/default/files/zensus_2011_gesamtdatensatz_svz.pdf.

[Kar] Bundesamt für Kartographie und Geodäsie. *Shape Files der Verwantungsgrenzen (UTM 32)*. URL: https://www.zensus2011.de/DE/Infothek/Begleitmaterial_Ergebnisse/Begleitmaterial_node.html.

[SC] National Institute of Statistics und Census of Costa Rica. *Sample of the Costa Rican census dataset, available from the microdata catalog of the national statistical office of Costa Rica—INEC—under a licensing agreement*. URL: http://%20sistemas.inec.cr/pad5/index.php/catalog/113.

[Sta16] Forschungsdatenzentren der Statistischen Ämter des Bundes und der Länder. *Metadatenreport Produkt 3 - Gesamtdatensatz Zensus 2011*. 2016. URL: https://www.forschungsdatenzentrum.de/sites/default/files/zensus_2011_gesamtdatensatz_mdr.pdf.

Data Access to Products of the Research Data Center

[Rot19] Patrick Rothe. *Statistische Geheimhaltung – Der Schutz vertraulicher Daten in der amtlichen Statistik*. 2019. URL: https://www.forschungsdatenzentrum.de/sites/default/files/fdz_aufsatz_1_gh.pdf.

[Züh+05] Dr. Sylvia Zühlke u. a. *Fachliche Informationen und Informationen zum Datenangebot zu dieser Veröffentlichung*. 2005. URL: `https://www.forschungsdatenzentrum.de/de/veroeffentlichungen/arbeitspapiere/3`.