IHSN SDC Introduction

Authors:

Matthias Templ, Bernhard Meindl and Alexander Kowarik
http://www.data-analysis.at

Vienna, November 4, 2013

Acknowledgement: International Household Survey Network (IHSN) and PARIS21/OECD

1 Concepts to protect microdata

A micro data file is defined as a data set consisting of observations on units. Units can be individuals, households or enterprises, for example. For each observation a set of variables is recorded and available in the data set. Within the framework of statistical disclosure control, it is helpful to divide the variables into different groups (see Section 1.1).

For the application of those methods, a workflow is presented in Section 1.4. This workflow shows how the process to anonymize micro data can possibly be performed. Subsequently, the concept of measuring associated disclosure risks given a disclosure scenario is discussed in Section 2. In Section 3, the main ideas of popular anonymisation methods are discussed.

1.1 Categorization of variables for statistical disclosure control

It is possible to classify variables into three groups that are however not necessarily disjunct.

- **Direct Identifiers** are variables that surely identify statistical units. For example, the social insurance numbers, names of companies or persons as well as addresses are considered as direct identifiers.

- **Key variables** can be seen as a set of variables that - if considered together - can be used to identify individual units. For example it might be possible to identify some individuals by using the combination of variables *gender*, *age*, *region* and *occupation*. Other examples for (confidential) key variables could be *income*, *health status*, *nationality* or *political preferences*. Often this key variables are also labeled as **implicit identifiers** or **quasi identifiers**. It has some advantages when discussing methods to distinguish between **categorical** and **continuous key variables** based on the scaled of the corresponding variables.

- **Non-confidential variables** are finally all variables that are not classified in any of the former two groups.

For specific methods (for example *l-diversity*), another group of **sensitive variables** can be defined.

1.2 What is disclosure?

In general, disclosure occurs if someone is able to learn something about someone that was not previously known using released data. The goal is to provide datasets for release that do not allow users to link information to specific units. It is possible to distinguish three different types of disclosure:

- **Identity disclosure**: In this case it is possible to associate an individual with a released data record that contains sensitive information. Disclosure is possible through direct identifiers, rare combinations of values in the key variables and exact knowledge on continuous key variables values in external data bases. For the latter case, extreme data values (for example extremly high turnover values for an enterprise) lead to high re-identification risks.

- **Attribute disclosure**: This means that it is possible to learn a characteristic of an individual based on information available in the released data. For example, if all people with *age = 56-60*, *race = black* and *region = 12345* have *labour status = unemployed*, this would lead to disclosure for variable *labour status*.
- **Inferential disclosure**: Identity- or attribute disclosure is derived on a statistical basis. In this case it is not possible to get exact matches, but for example matches with very high probability.

If linkage is successful based on such identifers, intruders have all information from the released data for the corresponding unit. This means, only a subset of (critical) variables can be exploited to learn everything about a unit that is available in the data set.

1.3 Remarks on SDC-methods

In gereral, methods for statistical disclosure control methods borrow a lot of techniques from other fields. For instance, multivariate (robust) statistics are used to modify or simulate continuous variables and to quantify information loss, distribution fitting methods are used to quantify disclosure-risks and statistical modelling methods form the basis of perturbation algorithms, to simulate data and to quantify risks and information loss. Linear programming ideas are used to modify data but keep the impact on data quality as low as possible.

The real problems and challenges are related to huge data sets and the need of efficient algorithms and implementations. Another layer of complexity is related to complex structures of hierarchical, multidimensional data sampled with complex survey-designs. Especially for computation time issues, missing values are a challenge but also structural zeros have great impact on the application of SDC-methods. Furthermore, the compositional nature of many components should always be considered and adds even more complexity. Statistical disclosure control techiques itself can be divided into three (broad) topics:

- measuring disclosure risk (see Section 2)
- perturbation and simulation of complex micro data (see Section 3)
- comparing original and modified data (information loss) (see Section 4)

1.4 Workflow

Figure 1 outlines the most common tasks, practices and steps of actions that are required to obtain confidential data. These steps are now motivated:

1. The first step is always be to remove all direct identification variables and variables that contain direct information on units from the microdata set.
2. Secondly, the key variables which for the foundation of all further risk-calculations have to be determined. This decision is subjective and often involves discussions with subject matter specialists and even the interpretation of the related (national) laws. Please, see Templ et al. [2014a] for practical applications on how to define key variables. For the simulation of fully synthetic data, the choice of key variables is not necessary since all variables produced synthetically, see for example [Alfons et al., 2011].

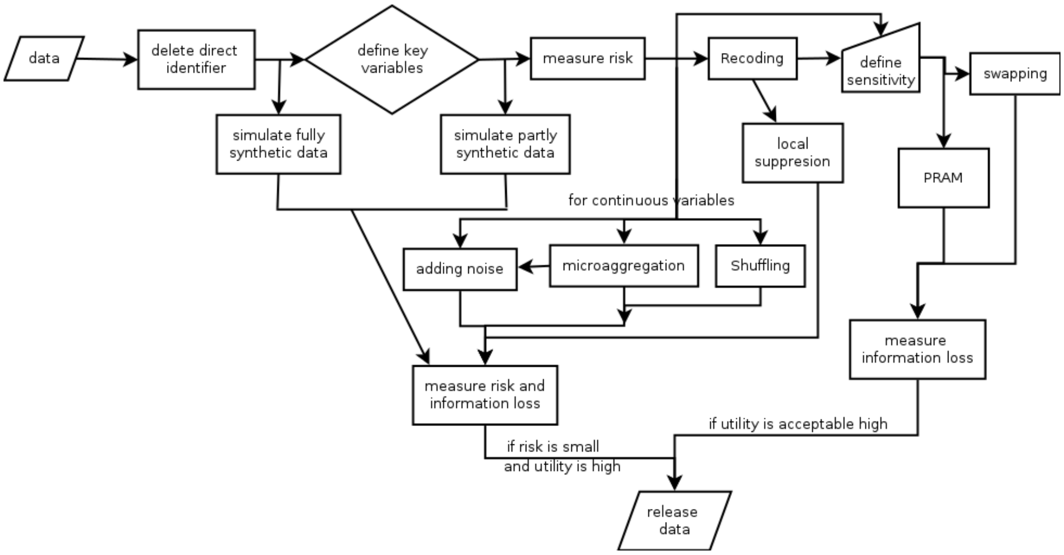

Figure 1: Possibilites for anonymising micro data using different SDC methods.

3. After the selection of key variables, disclosure risks of individual units must be measured. This includes the analysis of sample frequency counts as well as the application of probability methods to estimate the corresponding individual re-identification risks by taking population frequencies into account. Details on the underlying methodology are presented in Section 2.

4. Observations with high individual risks may then be modified. Techiques such as recoding and local suppression, recoding and swapping or post randomization (pram) can be applied to categorical key variables. In principle, pram or swapping can also be applied without prior recoding of key variables, however a lower swapping rate might be possible if recoding is applied before. The decision which method to apply also depends on the structure of the key variables. In general, one can use recoding together with local suppression if the amount of unique combinations of the key variables is low. Post-randomization should be used if the number of key variables is large and the number of unique combinations is high. For details, see Section 3.1, 3.3 and for practical applications Templ et al. [2014a].

   The values of continuously scaled key variables have to be perturbed as well. In this case, microaggragation is always a good choice (see Section 3.4). More sophisticated methods such as shuffling (see Section 3.6) also ofen lead to promising results.

5. After modifying categorical and numerical key variables of the microdata, information loss and the disclosure risk measures must be estimated. The ultimate goal is to release a safe microdata set that has low risk of linking confidential information to individuals and that still has high data utility. If the risks are low enough and the data utility is high, the anonymized data set is ready for release. If not, the entire anonymization process has to be repeated either with additional perturbations (if the remaining re-identification risks are too high) or with actions that will increase the data utility. For details on issues related to the dependency of both the utility and the risk, see Section 4 and Figure 2.

In general, the following recommendations hold:

Recommendation 1: Carefully choose the set of variables that are disclosive using knowledge of both subject matter experts and disclosure control experts.

Recommendation 2: Always perform a frequency- and risk estimation in order to evaluate how many observations have a high risk of dislosure given the selection of key variables.

Recommendation 3: Apply recodings to reduce uniqueness given the set of categorical key variables. This approach should be done in an exploratory manner. However, recodings on a variable should also be based on expert knowledge to combine categories that are reasonable to combine. Alternatively, swapping procedures may be applied on the categorical key variables so that data intruders can not be certain anymore if an observation has or has not been perturbed.

Recommendation 4: If recoding was applied, apply local suppression to achieve k-anonymity. In practice, parameter k is often set to 3.

Recommendation 5: Apply microaggregation to continuously scaled key variables. This automatically provides k-anonymity for these variables.

Recommendation 6: Quantify the data utility not only using typical estimates (like quantiles or correlations) but also on the most important data-specific benchmarking indicators.

Recoding and microaggregation works well to obtain non-confidential data with high data quality. While the disclosure risks cannot be calculated in a meaningful way if swapping methods like rank swapping or post-randomization have been applied, these methods have its advantages whenever a large number of key variables was selected. This is due to the fact that a high number of key variables leads to a high number of unique combinations that cannot be significantly reduced by applying recoding.

1.5 Risk versus data utility and information loss

As it was already mentioned before, the goal is always to release a safe microdata set that has low risk of linking confidential information to individual respondents and that still has high data utility. Figure 2 shows a typical situation. We applied the same method with different parameters to the Structural Earnings Statistics data [more on anonymisation on this data set can be found in Templ et al., 2014a]. For method 1, the parameter varies between 10 (small perturbation) to 100 (perturbation is ten times higher). When the parameter value was 100, the disclosure risk is low (since the data are heavily perturbed). But on the other hand the information loss is very high which also corresponds to very low data utility. Having a data set with only low perturbation applied, both the risk and the data utility is high. In any case the aim is to have a method that gives low risk and high data utility which is the region in the lower left area of the figure. It is easy to see that data anonymised with method 2 have considerable lower risk. Thus this method is preferable. Also the information loss increases only slightly if the parameter value increases. Thus method 2 with parameter value of approximately 7 would be a good choice in this case.

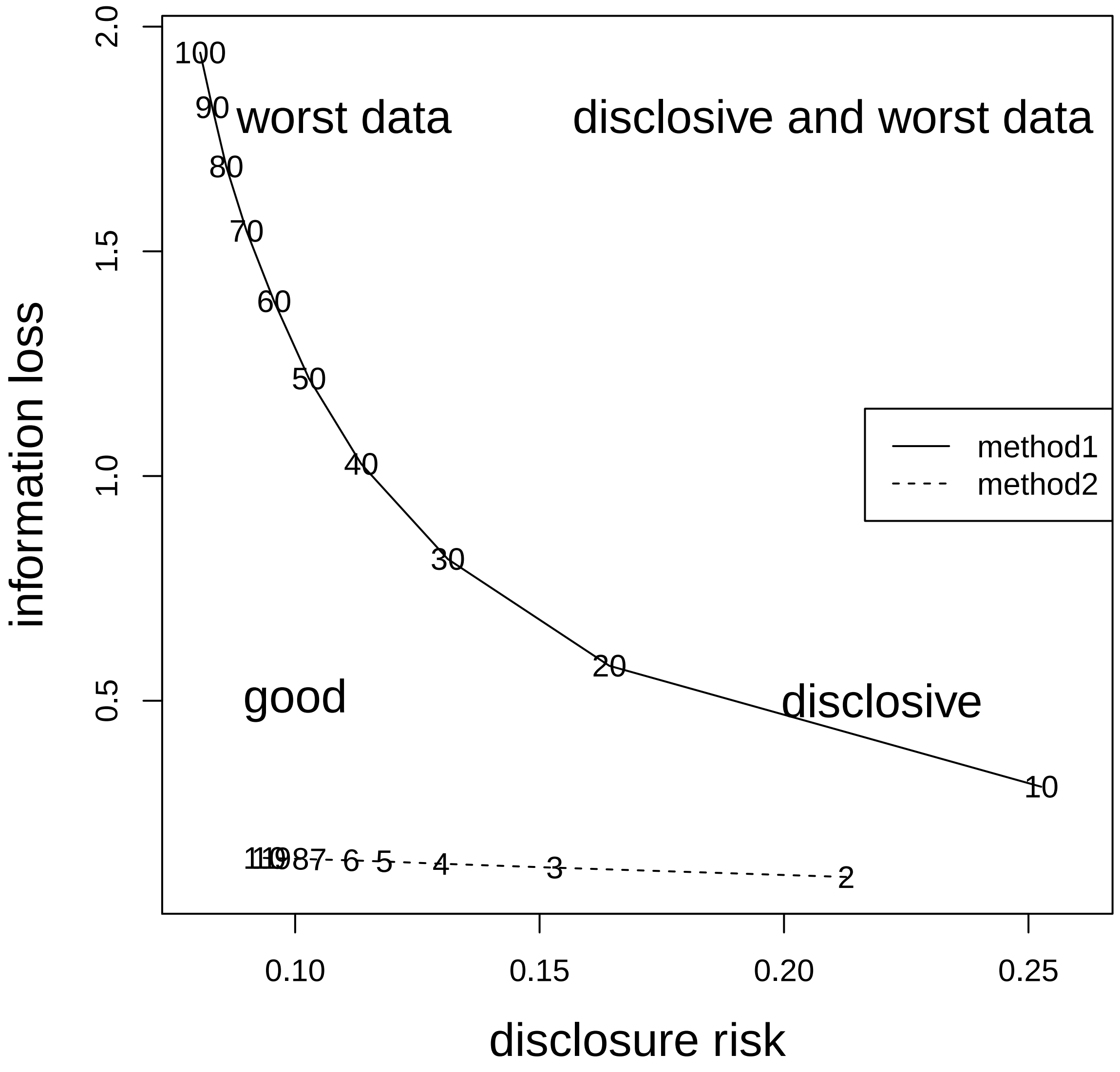

Figure 2: Risk versus information loss obtained for two specific perturbation methods and different parameter choices applied to SES data.

In real world examples things are often not as clear and thus data anonymization specialist have to make some decisions based on following considerations:

What is the legal situation regarding data privacy? Laws on privacy are varying between countries. Some countries have quite restricitive laws on data privacy, some not. Laws in one country are often different for different kind of data (business statistics, labor force statistics, social statistics, medical data, ...) which has to be taken into account.

How sensitive is the information in the data and who will be able to get access to the anonymized data file? Usually, laws consider two different kind of data users: users from universities and other research organizations or general users - the public. In the first case, special contracts are often made between data users and data holders. Usually, these contracts explicitly forbid the usage of the data aside very specific research projects and only allow to save the data within safe work environments. For these kind of users, anonymized micro data files are called *scientific use files*, whereas data for the public are typically referred to as *public use files*. Of course, the disclosure risk of a public use file needs to be very low and especially (much) lower as the corresponding risks remaining in scientific use files. For scientific use files, data utility is typically considerably higher than data utility of public use files.

Another aspect that must be considered is the sensitivity of the data set. Data about medical treatment of people are be more sensitive than turnover values and number of employees from establishments. If the data contains very sensitive information, the micro data should also be protected more than data that only contain information that is not likely to be attacked by data-intruders.

Which method is suitable for which purpose? The application of some specific methods results in low disclosure risk and large information loss, other methods may provide data with acceptable low disclosure risks. Other methods, like swapping or post randomization, may provide high or low disclosure risks and data utility depending on the specific choice of parameter values.

In any case, data holders should always estimate the disclosure risk for their original data sets as well as the disclosure risks and data utility for anonymized versions of the data. To achieve good results (low disclosure risk, high data utility), it is necessary to do the anonymisation in an explanatory manner by trying to apply different methods using different parameter settings until the a suitable trade-off between risk and data-utility has been achieved.

2 Measuring the Disclosure Risk

Measuring risk in an micro data set is a key task. Risks are essential to be able to decide if the data set is protected enough to be released. To assess the disclosure risk, it is required to make realistic assumptions on the information data users might have at hand to match against the micro data set. These assumptions are called *disclosure risk scenarios*. Based on a specific disclosure risk scenario, it is necessary to define a set of key variables (identifying variables) that can be used as input for the risk evaluation procedure.

Typically, risk evaluation is based on the concept of *rareness* or rather *uniqueness* in the sample and/or in the population. The interest is on individual units that possess rare combinations the the selected key variables. The basic assumtion is that units having rare combinations of key variables can be identified easier and thus have higher risk of re-identification. It is possible to cross tabulate all identifying variables and have a look at its cast. Patterns[1] that only very few individuals possess are in this sense considered as risky - especially if these observations also have small sampling weights. This means that the expected number of individuals with the this very pattern is expected to be low in the population as well.

[1] a pattern is defined as a specific combination of the values of all key variables

2.1 Frequencies Counts

Let us define frequency counts also in a mathematical notation. Consider a random sample of size n drawn from a finite population of size N. Let $\pi_j,\ j = 1, \ldots, N$ be the (first order) inclusion probabilities, the probability that element u_j of a population of the size N is chosen in a sample of size n.

All possible combinations of categories in the key variables (for short: *keys* or *patterns*) can be calculated by cross tabulation of these variables. Let $f_i,\ i = 1, \ldots, n$ be the frequency counts obtained by cross tabulation and let F_i be the frequency counts of the population which belong to the same pattern. If $f_i = 1$ applies, the corresponding observation is unique in the sample given the key-variables. If $F_i = 1$, then the observation is unique in the population as well and (automatically) also unique in the sample.

F_i is usually not known since in statistics usually information on samples is collected and only few information about the population is known from registers and/or other external sources and therefore has to be extimated. Basic estimation of F_i is shown in Table 1. In Section 2.5 it is shown how to deal with population frequency counts and how to estimate the frequency counts in a more realistic way. In Table 1 a very simple data set is used to explain the calulation of sample and population frequency counts. One can easily see that observation 1 and 8 are equal given the key-variables *Key1*, *Key2*, *Key3* and *Key4*. The values of observations 1 and 8 are equal and therefore the sample frequency counts are $f_1 = 2$ and $f_8 = 2$. The estimated population frequencies are obtained by summing up the sample weights for equal observations. Population frequencies $\hat{F}_1$ and $\hat{F}_8$ can then be estimated by summation over the corresponding sampling weights, w_1 and w_8. In summary, two observations with the pattern $(1, 2, 5, 1)$ exist in the sample and 110 observations with these pattern can be expected to exist in the population.

Table 1: Example of sample and estimated population frequency counts.

	Key1	Key2	Key3	Key4	w	risk	fk	Fk
1	1	2	5	1	18.0	0.017	2	110.0
2	1	2	1	1	45.5	0.022	2	84.5
3	1	2	1	1	39.0	0.022	2	84.5
4	3	3	1	5	17.0	0.177	1	17.0
5	4	3	1	4	541.0	0.012	1	541.0
6	4	3	1	1	8.0	0.297	1	8.0
7	6	2	1	5	5.0	0.402	1	5.0
8	1	2	5	1	92.0	0.017	2	110.0

When using the graphical user interface of `sdcMicro` - the `sdcMicroGUI` - the frequencies are updated as soon categorical key variables are modified.

2.2 The concept of k-anonymity

Based on a set of key variables, a desired characteristic of a protected microdata set is often to achieve k-anonymity [Samarati and Sweeney, 1998, Sweeney, 2002]. This means that each possible pattern of key variables contains at least k units in the micro data. This is equal to $f_i \geq k\ , i = 1, ..., n$. A typical value is $k = 3$.

Table 2: k-anonymity and l-diversity on a toy data set.

	key1	key2	sens	fk	ldiv
1	1	1	50	3	2
2	1	1	50	3	2
3	1	1	42	3	2
4	1	2	42	1	1
5	2	2	62	2	1
6	2	2	62	2	1

k-anonymity is typically achieved by recoding categorical key variables into less categories (see Section 3.1) and by additionally suppressing specific values in the key variables of some units. For details see Section 3.2.

For local suppression, function **localSuppression**() of the `R` package `sdcMicro` can be used to accomplish k-anonymity [for details have a look in Templ et al., 2014b, 2013]. In this implementation a heuristic algorithm is called to suppress as few values as possible. It is possible to specify a desired ordering of key variables in terms of importance which the algorithm takes into account. It is even possible to specify key variables that are considered of such importance that almost no values for these variables are suppressed. This function can also be used in the graphical user interface of the `sdcMicroGUI` package [Kowarik et al., 2013, Templ et al., 2014b].

2.3 l-Diversity

An extension of k-anonymity is l-diversity [Machanavajjhala et al., 2007]. Consider a group of observations with the same pattern in the key variables and let the group fulfill k-anonymity. A data intruder can therefore by definition not identify an individual of this group. However, if all observations have the same entries in an additional sensitive variable (for example *cancer* in the variable *medical diagnosis*), an attack is successful if the attacker can identify at least one individual of the group. The attacker knows that this individual has cancer with certainty. The distribution of the target sensitive variable is referred to as l-diversity.

In Table 2 we consider a small example data set that highlights the calculations related to calculate l-diversity It also points out the (slight) difference compared to k-anonymity. The first two columns present the categorical key variables. The third column of the data defines a variable containing sensitive information. Sample frequency counts f_i are printed in the fourth column. They equal 3 for the first three observations, the fourth observation is unique and for the last two observations, the frequency counts f_i are 2. Only observation four violates 2-anonymity. Looking closer at the first three observations we see that only two different values are present in the sensitive variable. Thus the l-(distinct)-diversity is just 2. For the last two observations, 2-anonymity is achieved but still the intruder knows the exact information of the sensitive variable. For these observations the l-diversity-measure is 1 indicating that the sensitive information can be disclosed, since the value of the sensitive variable is $= 62$ for both these observations.

Differences in values of the sensitive variable can be measured differently. We presented here the distinct diversity that counts how many different values exist within a pattern. Additional methods (entropy, recursive, multi-recursive) are implemented in the software, for more information see the help files of `sdcMicro`.

2.4 Sample frequencies on subsets: SUDA

SUDA (**S**pecial **U**niques **D**etection **A**lgorithm) estimates disclosure risks for each unit. SUDA2 [see, e.g., Manning et al., 2008] is a recursive algorithm to find minimal sample uniques. The algorithm generates all possible variable subsets of selected (categorical) key variables and scans for unique patterns within subsets of these variables. The risk of an observation finally depends on two aspects.

(a) The lower the amount of variables needed to receive uniqueness, the higher the risk (and the higher the *suda score*) of the corresponding observation.

(b) The larger the number of minimal sample uniquenes contained within an observation, the higher the risk of this observation.

(a) is calculated for each observation i by $l_i = \prod_{k=MSUmin_i}^{m-1} (m-k) \quad , i = 1, ..., n$. In this formula, m corresponds to the *depth*, the maximum size of variable subsets of the key variables, $MSUmin_i$ the number of minimal uniques of observation i and n the number of observations of the data set. Since each observation is treated independently, a specific value l_i belonging to a specific pattern are summed up. This results in a common suda score for each of the observation that are contained in this pattern (this summation is the contribution of (b)).

The final SUDA score is then calculated by normalizing these suda score by dividing them by $p!$, with p being the number of key variables. To receive the so called DIS score - loosely speaking - an iterative algorithm based on sampling of the data and matching of subsets of the sampled data with the original data is applied. In this algorithm the probabilities of correct matches given unique matches are calculated. It is however out of scope to exactly describe this algorithm here. Thus we refer to Elliot [2000] for details. The DIS suda score is calculated from the suda and the DIS scores and is available in `sdcMicro` as `disScore`).

Note that this method does also not consider population frequencies in general but consider sample frequencies on subsets. The DIS suda scores somehow approximately consider based on the sample information population uniqueness by simulation, but - to our knowledge - in generally it do not consider sampling weights and biased estimates may therefore result.

Table 3: Example of suda scores (scores) and dis suda scores (disScores).

	Key1	Key2	Key3	Key4	fk	scores	disScores
1	1	2	5	1	2	0.00	0.0000
2	1	2	1	1	2	0.00	0.0000
3	1	2	1	1	2	0.00	0.0000
4	3	3	1	5	1	3.50	0.0164
5	4	3	1	4	1	0.00	0.0000
6	4	3	1	1	1	0.00	0.0000
7	6	2	1	5	1	1.75	0.0072
8	1	2	5	1	2	0.00	0.0000

In Table 3 we use the same test data set as in Section 2.1. Also the sample frequency counts f_i but also the suda and DIS suda scores have been calculated.

The suda scores have the largest value for observation 4 since also subsets of key variables of this observation are unique, while for observations $1-3$,$5-6$ and 8, no subset is unique.

Suda or rather suda2 [SUDA2, Manning et al., 2008] is implemented in `sdcMicro` as function **suda2**() based on `C++` code from the IHSN. Additional output such as the contribution percentages of each variable to the score are also available as output of this function. The contribution to the suda score is calculated by looking how often a category of a key variable contributes to the score.

2.5 Population frequencies and the individual risk approach

To assess if an unit is at risk, typically a threshold approach is used. If the individual risk of re-identification for an individual is above a certain threshold value, the unit is said to be at risk. To compute individual risks, it is necessary to estimate the frequency of a given key (pattern) in the population. In the previous section, Section 2.1, the population frequencies have already been estimated. However, one can show that these estimates almost always overestimate small population frequency counts [see also Templ and Meindl, 2010] and should not be used to estimate the disclosure risk.

A better approach is to use so-called super-population models. In such models population frequency counts are modeled given certain distribution. The estimation procedure of sample counts given the population counts can be modeled for example by assuming a negative binomial distribution [see Rinott and Shlomo, 2006] and is implemented in `sdcMicro` in function measure_risk() [for details, see Templ and Meindl, 2010]. Of course, this calculations can also be done from within the graphical user interface.

In Table 4 all concepts that were now discussed have been applied. The estimation of frequency counts for the sample and on population level which corresponds to the sum of sampling-weights for each group, the l-diversity measure, the suda algorithm and the individual risk estimation are listed in this table. One can observe that the individual risk is low for observation 5 since the sampling weight of this unit is quite high. Thus, one can assume that this observation is not likely to be unique in the population. On the other hand, the individual risk of observations that are sample uniques ($f_i = 1$) in combination with small sampling weights is relatively high. This means that the inclusion probability of each individual is taken into account when estimating the individual risks.

Table 4: Display of frequency counts, l-diversity, suda and individual risk. The continuous variable (Num3) was chosen as sensitive variable for l-diversity.

	Key1	Key2	Key3	Key4	Num3	w	fk	Fk	ldiv	suda	risk
1	1	2	5	1	4	18.0	2	110.0	2	0.00	0.0171
2	1	2	1	1	22	45.5	2	84.5	2	0.00	0.0220
3	1	2	1	1	8	39.0	2	84.5	2	0.00	0.0220
4	3	3	1	5	91	17.0	1	17.0	1	2.25	0.1771
5	4	3	1	4	13	541.0	1	541.0	1	1.75	0.0117
6	4	3	1	1	14	8.0	1	8.0	1	1.00	0.2971
7	6	2	1	5	1	5.0	1	5.0	1	2.25	0.4024
8	1	2	5	1	5	92.0	2	110.0	2	0.00	0.0171

2.6 Calculating cluster (household) risks

Micro data sets often contain hierarchical cluster structures. For social surveys this is for example often the case, if individuals are clustered in households. The risk of re-identification of an individual within an household may also have an affect on the probability to disclose other members in the same household. Thus, the household or more generally cluster-structure of the data must be taken into account when calculating risks.

It is commonly assumed that the risk of reidentification of a household is the risk that at least one member of the household can be disclosed. Thus this probability can be simply estimated from individual risks as 1 minus the probability that no member of the household can be identified. This is also the implementation strategy from `sdcMicro`.

2.7 Measuring the global risk

In Sections 2.5 and 2.6 the theory of individual risks and the extension of this approch to clusters such as households was discussed. However, in a lot of applications it is desired to estimate a measure of the global risk. Any global risk measure will result in one single number that that can be used to assess the risk of an entire micro data set.

2.7.1 Measuring the global risk using individual risks

Two approaches are listed here which may be used to determine the global risk for a data set using individual risks:

- **Benchmark:** In this approach it is counted how many observation can be considered risky and also have higher risk as the main part of the data. For example we consider units with individual risks being ≥ 0.1 and twice as large as the median of all individual risks + $2 \cdot$ MAD of all unit risks. The MAD is defined as the median absolute deviation.

- **Global risk:** the sum over the individual risks in the data set gives the expected number of re-identifications [see also Hundepool et al., 2008].

The benchmark-approach gives an indication if the distribution of individual risk occurs contains extreme values. It is an relative measure that depends on the distribution of individual risks. It is not valid to conclude that observations with higher risk as this benchmark are of very high risk. It evaluates if some unit risks behave differently compared to the main bulk of the individual risks, which is anyhow an interesting and helpful information. The second approach is based on an absolute measure of risk. The print output of the corresponding function from `sdcMicro` shows both measures and is printed below:

```
--------------------------
0 obs. with higher risk than the main part
Expected no. of re-identifications:
 0.97 [ 12.08 %]
--------------------------
```

If a cluster (for example a household ID) has been defined, then also a global risk measure that takes into account this hierarchical structure is reported.

2.7.2 Measuring the risk using log-linear models

Sample frequencies, considered for each of M patterns m, f_m , $m = 1, ..., M$ can be modeled by a poisson distribution. In this case the global risk can be defined as [see Skinner and Holmes, 1998]

$$\tau_1 = \sum_{m=1}^{M} \exp\left(-\frac{\mu_m(1-\pi_m)}{\pi_m}\right), \qquad \text{with } \mu_m = \pi_m\lambda_m. \tag{1}$$

For simplicity, the (first order) inclusion probabilities are assumed to be equal, $\pi_m = \pi$, $m = 1, ..., M$. τ_1 can be estimated by log-linear models that include both the main effects and possible interactions. The model is defined as:

$$\log(\pi_m\lambda_m) = \log(\mu_m) = \mathbf{x}_m\beta.$$

To estimate the μ_m's, the regression coefficients β have to be estimated, for example using iterative proportional fitting. The quality of this risk measurement approach depends on the number of different keys (that result from cross tabulation all key variables). If the cross tabulated key variables are sparse in terms of how many observations have the same patterns, predicted values might be of low quality. It also has to be considered that if the model for prediction is weak, also the quality of the prediction of the frequency counts is weak. Thus, the risk measurement with log-linear models might only lead to acceptable estimates of the global risk if not too many key variables are selected if good predictors are available in the data set.

In `sdcMicro` the global risk measurement using log-linear models can be done with function **LLmodGlobalRisk**(). However, this function is experimental and should only be used by expert users and still needs further testing.

2.8 Measuring risk for continuous key variables

The concepts of uniqueness and k-anonymity can not directly be applied to quantitative variables. The reason is that a direct application would give as result that (almost) every unit in the data set is unique given a set of continous key variables. Hence, this approach will fail. In the following sections methods to measure risk for continous key variables are presented.

2.8.1 Problem of linking information

If detailed information about a value of a numerical variable is available, attackers may be able to identify and eventually obtain further information about an individual. Thus, an intruder may be able to identify statistical units by applying for example linking or matching algorithms. The anonymization of continuous key variables should avoid the possibility of successfully merging the underlying micro data with other (external) data sources.

We assume that an intruder has information about a statistical unit which is included in the micro data and the information of the intruder overlaps on some variables with the information in the data. In simpler terms we assume that the intruder's information can be merged with the micro data that should be protected. In addition to that we also assume that the intruder is sure that the link to the data is correct, except for microaggregated data (see Section 3.4). In this case, an intruder can not be sure if the link is valid because at least k observations have the same value for each continuous variable.

2.8.2 Distance-based record linkage

The underlying idea of distance based record linkage methods is to find the nearest neighbours between observations of two different data sets. Domingo-Ferrer and Torra [2001] showed that these methods outperform probabilistic methods. Such probabilistic methods are often based on the EM-algorithm which is highly influenced by outliers.

Mateo-Sanz et al. [2004] introduced distance based record linkage and interval disclosure. In the first approach they look for the nearest neighbor from each observation of the masked data value to the original data points. Then they mark those units for which the nearest neighbor is the corresponding original value. In the second approach they check if the original value falls within an interval which is centered around the masked value. Then they calculate the length of the intervals based on the standard deviation of the variable under consideration (see also Figure 3, left upper graphic).

2.8.3 Special treatment of outliers when calculating disclosure risks

Almost all data sets used in official statistics contain units whose values in at least one variable are quite different from the main part of the observations. This leads to the fact that these variables are very asymmetric distributed. Such outliers might for example be enterprises with a very high value for turnover or persons with extremely high income. Also multivariate outliers exist [see, for example Templ and Meindl, 2008a].

Unfortunately, intruders can have an interest to disclose a large enterprise or an enterprise which has specific characteristics. Since enterprises are often sampled with certainty or have a sampling weights close to 1, intruders can often be very confident that the enterprise they wants to disclose has definitely been sampled. In contrast to that, an intruder may not be as interested to disclose statistical units which exhibit the same behavior as the main part of the other observations in variables of interest. For these reasons it is good practice to define measures of disclosure risk that take the *outlyingness* of an observation into account. For details refer to Templ and Meindl [2008a]. The key idea is to assume that outliers should be much more perturbed than non-outliers because these units are easier to re-identify even when the distance from the masked observation to its original observation is relatively large.

This method for risk estimation (in Figure 3 called RMDID2) is also included in the `sdcMicro` package and works as described in Templ and Meindl [2008a] and listed below:

1. Robust mahalanobis distances (RMD) [see, for example Maronna et al., 2006] are estimated in order to obtain a robust, multivariate distance for each unit.

2. Intervals are estimated for each observation around every data point of the original data points. The length of the intervals depend on squared distances calculated in step 1 and an additional scale parameter. The higher the RMD of an observation, the larger are the corresponding intervals.

3. Check if the corresponding masked values of a unit fall into the intervals around the original values or not. If the masked value of an observation lies within such an interval, the entire observation is considered unsafe. We obtain a vector indicating which observations are safe or which are not.

4. For all unsafe units it is checked if at least m other observations from the masked data are very close. The quantification of *close* is done by specifying a parameter for the length of the intervals around this observation using euclidean distances. If more than m points lie within these small intervals we can conclude that the observation is *safe*.

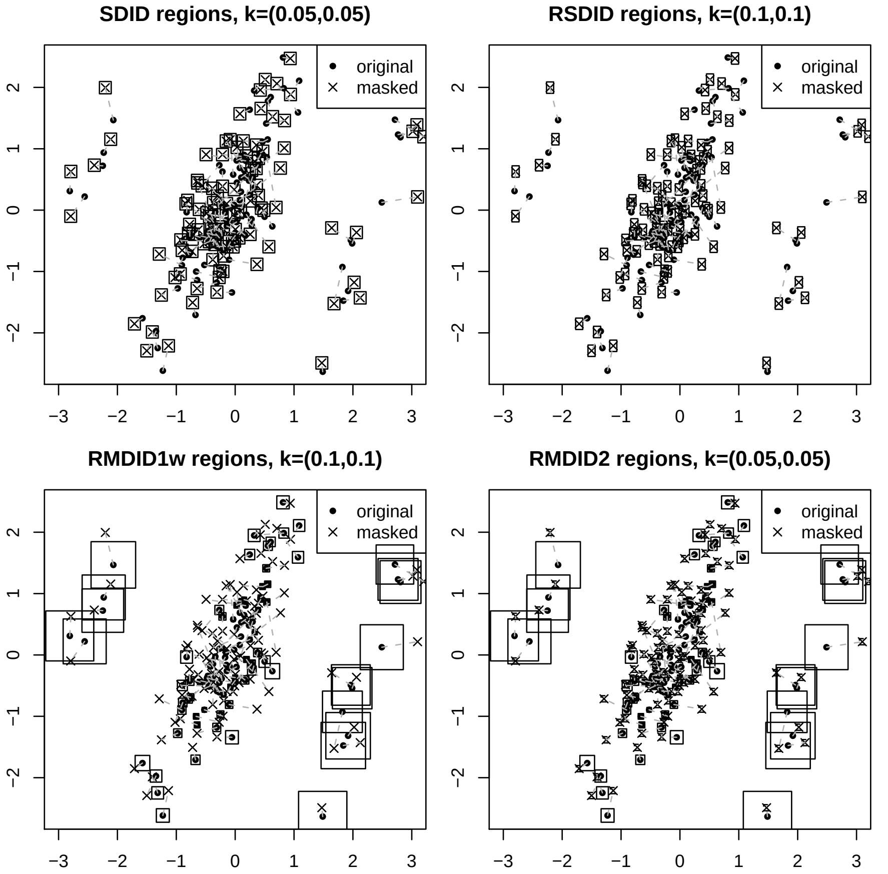

Figure 3: Original and corresponding masked observations (perturbed by adding additive noise). In the bottom right graphic small additional regions are plotted around the masked values for *RMDID2* procedure.

Figure 3 points out the idea of weighting the disclosure risk intervals. For simple methods (left and right graphics on the top) the rectangular regions around each value are the same size for each observation. Our proposed methods takes the robust mahalanobis distances of each observation into account. The difference between the bottom right and the bottom left graphic is that for method *RMDID2* rectangular regions are calculated around each masked variable as well. If an observation of the masked variable falls into an interval around the original value, it is checked if this observation does have close neighbours. If the values of at least m other masked observations can be found inside a second interval around this masked observation. These observations are considered as *safe*.

The methods just discussed are also implemented and available in `sdcMicro` as functions **dRisk**() and **dRiskRMD**(). The former is automatically applied to objects of class *sdcMicroObj*, while the latter has to be specified explicitly.

3 Anonymisation Methods

In general, two different kinds of anonymisation methods can be defined: deterministic and probabilistic methods. For categorical variables, recoding and local suppression are deterministic procedures while swapping and post randomization [Gouweleeuw et al., 1998] are based on randomness are considered as probabilisitic methods. For continuous variables, microaggregation is a deterministic method while adding (correlated) noise [Brand, 2004] and shuffling [Muralidhar et al., 1999] are probabilistic procedures. Whenever probabilistic methods are applied, the random seed of the pseudo random number generator of the software to ensure reproducibility of the results should be fixed.

3.1 Recoding

(global) Recoding is a non-perturbative method that can be applied to both categorical and continuous key variables. The basic idea of recoding a categorical variable is to combine several categories into a new, less informative category. If the method is applied to a continuous variable it means to discretize the variable. The main concept in both cases is to reduce the total number of possible outcomes of a variable. Typically, recoding is applied to categorical variables where the number of categories with only few observations (extreme categories) is reduced.

A special case of global recoding is *top and bottom coding* which can be applied to ordinal and categorical variables. The main idea for this approach is that all values above (top-coding) and/or below (bottom-coding) a pre-specified threshold values are combined into a new category. Function **globalRecode**() can be applied in sdcMicro to perform global recoding and also top/bottom coding. The help file with some examples is accessible using ?**globalRecode**. Note, that a more user-friendly version of global recoding can be applied using sdcMicroGUI.

3.2 Local Suppression

Local suppression is a non-perturbative method that is typically applied to categorical variables. The idea is to suppress certain values in at least one variable. Typically, the input variables are part of the set of key variables that are also used for the calculation of (individual) risks as it described in 2. Individual values are suppressed in a way that the set of variables agreeing on a specific pattern are increased. Local suppression is often used to achieve k-Anonymity as it was described in Section 2.2.

Using function **localSupp**() of sdcMicro it is possible to suppress the values of a key variable for all those units having individual risks given a disclosure risk scenario is above a pre-defined threshold. This procedure requires user intervention by setting the threshold. To automatically suppress a minimum amount of values in the key variables to achieve k-anonymity it is possible to use function **localSuppression**(). This algorithm also allows to specify a user-dependent preference that determines which key variables should be preferred when choosing values that need to be suppressed.

Specifying the importance of variables as an parameter in function **localSuppression** () allows to give key variables such high importance that suppressions will only take place if no other choices are possible. This is useful if - for example - an scientific use file with specific requirements needs to be produced. Still, it is possible to achieve k-anonymity for the selected key variables in almost all use-cases.

3.3 Post-randomization

Post-randomization (also referred to as PRAM) [Gouweleeuw et al., 1998] is a perturbation, probabilistic method that can be applied to categorical variables. The key idea is that the values of a categorical variable in the original microdata file are changed into other categories taking into account pre-defined transition probabilities. This process is usually modeled using a known transition matrix. Such a matrix lists for each category of a categorical variable probabilities to change into other possible categories.

A example would be to have a variable with only 3 categories, A1, A2 and A3. The transition of a value from category A1 to category A1 is, for example, fixed with probability $p_1 = 0.85$ which means that only with probability $p_1 = 0.15$ a value of A1 is changed to either A2 to A3. The probability of a change from categoriy A1 to A2 might be fixed with probability $p_2 = 0.1$ and changes from A1 to A3 with $p_3 = 0.05$. Also probabilities to change values from class A2 to the other classes and for A3, respectively, have to be specified beforehand. All transition probabilities have to be stored in a a matrix which is the main input to function **pram**() in `sdcMicro`. This example now applied, however the default parameters of **pram**() and not a custom transition matrix is used. We can observe from the following output that exactly one value changed the category. One observation having A3 in the original data has value A1 in the masked data.

```
> set.seed(1234)
> A <- as.factor(rep(c("A1","A2","A3"), each=5))
> A

 [1] A1 A1 A1 A1 A1 A2 A2 A2 A2 A2 A3 A3 A3 A3 A3
Levels: A1 A2 A3
```

We apply **pram**() on vector `A` and print the result:

```
> Apramed <- pram(A)
> Apramed

 this vector is perturbed with
 invariant PRAM

 Parameters for PRAM:
 alpha =  0.5
 minimum diagonal element =  0.8
```

The summary gives more detailed information. It shows a table of original frequencies as well as the corresponding table after the post-randomization procedure. Finally, also all the transitions that took place are listed:

```
> summary(Apramed)

 ----------------------
 original frequencies:

A1 A2 A3
 5  5  5
 ----------------------
 frequencies after perturbation:

A1 A2 A3
 6  5  4

 ----------------------
 transitions:
  transition Frequency
1    1 --> 1         5
2    2 --> 2         5
3    3 --> 1         1
4    3 --> 3         4
```

PRAM is applied to each observation independently and the procedure is random. This means that different solutions are obtained for every run of *PRAM* if no seed is specified for the random number generator. A main advantage of the post-randomization procedure is the flexibility of the method. Since the transition matrix can be specified freely as a function parameter, all desired effects can be modeled. For example it is possible to prohibit changes from one category to another by setting the corresponding probability in the transition matrix to 0.

Iin `sdcMicro`, **pram_strat**() allows to perform post-randomization. The corresponding help file can be accessed by typing ?**pram** into an `R` console or by using the help-menu of `sdcMicroGUI`. When using **pram_strat**() it is possible to apply pram to sub-groups of the micro data set, independently. In this case the user has to select the stratification variable defining the sub-groups. If the specification of this variable is omitted, the post-randomization procedure is applied to all observations in the data set.

3.4 Microaggregation

Microaggregation is a perturbative method that is typically applied to continuous variables. The main idea is that records are partitioned into groups. Within each group, the values of each variable are aggregated. Typically the arithmetic mean is used to aggregate the values, however other (robust) methods are also possible. Individual values of the records for each variables are replaced by the group aggregation value (often the mean), see for example Table 5. In this example always two values that are most similar are replaced by their column-wise means.

Depending on the method chosen in function **microaggregation**(), additional parameters can be specified. It is possible for example to specify the number of observations that should be aggregated as well as the statistic used to calculate the aggregation. This statistics defaults to be the arithmetic mean. It is also possible to perform microaggregation independently to pre-defined clusters or to use cluster methods to achieve the grouping.

All of the above settings (and many more) can be applied in `sdcMicro` using function **microaggregation**(). The corresponding help file can be viewed with command ?**microaggregation** or using the help-menu in `sdcMicroGUI`.

Table 5: Example of microaggregation. Columns 1-3 contain the original variables, columns 4-6 the microaggregated values.

	Num1	Num2	Num3	Mic1	Mic2	Mic3
1	0.30	0.400	4	0.65	0.85	8.5
2	0.12	0.220	22	0.15	0.51	15.0
3	0.18	0.800	8	0.15	0.51	15.0
4	1.90	9.000	91	1.45	5.20	52.5
5	1.00	1.300	13	0.65	0.85	8.5
6	1.00	1.400	14	1.45	5.20	52.5
7	0.10	0.010	1	0.12	0.26	3.0
8	0.15	0.500	5	0.12	0.26	3.0

3.5 Adding noise

Adding noise is a perturbative protection method for micro data that is typically applied to continuous variables. The main idea is to add statistical noise to continuous variables. This approach protects data against exact matching with external files if information on specific variables is available for example from registers.

While this approach sounds simple in principle, a lot of different algorithms can be used to overlay data with stochastic noise. It is possible to add uncorrelated random noise. In this case the noise is typically normally distributed and the variance of the noise term is proportional to the variance of the original data vector. Adding uncorrelated noise preserves means but variances and correlation coefficients between variables are not preserved. This statistical property is however respected if correlated noise method(s) are applied.

For the correlated noise method [Brand, 2004], the noise term is derived from a distribution having a covariance matrix that is proportional to the covariance matrix of the original micro data. In the case of correlated noise addition, correlation coefficients are preserved and at least the covariance matrix can be consistently estimated from the perturbed data. However, the data structure may differ a lot if the assumption of normality is violated. Since this is virtually always the case when working with real-world data sets, a robust version of the correlated noise method is included in `sdcMicro`. This method which allows departures from model assumptions is described in detail in Templ and Meindl [2008b]). More information cal also be found in the help file of the package by calling ?**addNoise** or using the help menu of the graphical user interface.

In `sdcMicro` several other algorithms are implemented that can be used to add noise to continuous variables. For example it is possible to add noise only to outlying observations. In this case it is assumed that such observations possess higher risks than non-outlying observations. Other methods make sure that the amount of noise that should be added takes the underlying sample sizes (or rather sampling weights) into account. Noise can be added to variables in `sdcMicro` using function **addNoise**() or using `sdcMicroGUI`.

3.6 Shuffling

Various masking techniques that are based on linear models have been developed in literature, such as multiple imputation [Rubin, 1993], general additive data perturbation [Muralidhar et al., 1999] and the information preserving statistical obfuscation synthetic data generators [Burridge, 2003]. These methods are capable of maintaining linear relationships between variables but fail to maintain marginal distributions or non-linear relationships between variables.

The main idea of shuffling [Muralidhar and Sarathy, 2006] is to simulate a synthetic values of the continuous key variables conditioned on independent non-confidential variables. After the simulation of the new values for the continuous key variables, reverse mapping (shuffling) is applied. This means that ranked values of the simulated values are replaced by the ranked values of the original data (columnwise). For more details on the method we refer to [Muralidhar and Sarathy, 2006]. In the implementation of `sdcMicro`, a model of almost any form and complexity can be specified, see ?**shuffling** for details.

4 Measuring data utility

It is of great interest to measure data utility of the micro data set after disclosure limitation methods have been applied and to assess the impact of these methods.

4.1 General applicable methods

Anonymized data should have the same structure of the original data and should allow any analysis with high precision.

To evaluate the precision, estimation of various classical estimates such as means and covariances are important. Using function **dUtility**() it is possible to calculate different measures based on classical or robust distances for continuous scaled variables. Estimates are computed for both the original and the perturbed data and are then compared. Three important information loss measures are now given:

- **IL1s** is a measures introduced by [Mateo-Sanz et al., 2004]. The this measure is given as $IL1 = \frac{1}{p}\sum_{j=1}^{p}\sum_{i=1}^{n}\frac{|x_{ij}-x_{ij}^{'}|}{\sqrt{2}S_j}$ and can be interpreted as scaled distances between original and perturbed values for all p continuous key variables.

- **eig** is a measure calculating relative absolute differences between eigenvalues of the covariances from standardized continuous key variables of the original and the perturbed variables. Eigenvalues can be estimated from a robust or classical version of the covariance matrix.

- **lm** is a measure based on regression models. It is defined as $|(\bar{\hat{y}}_w^o - \bar{\hat{y}}_w^m)/\bar{\hat{y}}_w^o|$, with $\bar{\hat{y}}_w$ being fitted values from a pre-specified model obtained from the original (index o) and the modified data (index m). Index w indicates that the survey weights should be considered when fitting the model.

Note that these measures are automatically estimated in `sdcMicro` when an object of class *sdcMicroOjb* is generated or whenever continuous key varibles are modifed in such an object. Thus, no user input is required.

4.2 Specific tools

In practice it is not possible to create an anonymized file with the very same structure as the original file. However, an important goal should always be that the difference of results of the **most important statistics** based on anonymized and original data should be very small or even zero. Thus, the idea is to measure the data utility based on benchmarking indicators [Ichim and Franconi, 2010, Templ, 2011a] which is in general a better approach to assess data quality than applying general tools.

The first step in quality assessment is to decide on a set of benchmarking indicators. In order to do so, one has to evaluate what users of the underlying data are analyzing and then try to decide on the most important estimates. These estimators are often called *benchmarking indicators* [see, e.g., Templ, 2011b,a]. Special emphasis should be put on benchmarking indicators that take into account the most important variables of the micro data set. Also, indicators that refer to the most sensitive variables within the micro data should be calculated. The general procedure is quite simple and can be described in the following steps:

- Selection of a set of benchmarking indicators
- Choice of a set of criteria on how to compare the indicators
- Calculation of all benchmarking indicators on the original micro data
- Calculation of the benchmarking indicators on the protected micro data set
- Comparison of statistical properties such as point estimates, variances or overlaps in confidence intervals for each benchmarking indicator
- Assessment if the data utility of the protected micro data set is good enough to be used by researchers

If the quality assessment in the last step of the sketched algorithm is satisfactory, the anonymized micro data set is ready to be published. If the deviations of the main indicators calculated from the original and the protected data are too large, the anonymization procedure should be restarted and modified. It is possible to either change (some) parameters of the procedures that have been applied or to start from scratch and completely change the anonymization process.

Usually the evaluation is focused on the properties of numeric variables given unmodified and modified micro data. However, it is of course also possible to have a look at the impact of local suppression or recoding that has been conducted to reduce individual re-identification risks. Another possibility to evaluate the data utility of numerical variables is to define a model that is fitted on the original, unmodified microdata. The main idea is to predict important, sensitive variables using this model both for the original and the protected micro data set in a first step. In a second step, statistical properties of the model results, such as the differences in point-estimates or variances, are compared for the predictions given original and modified micro data and the resulting quality is assessed. If the deviations are small enough one may go on to publish the safe and protected micro data set. Otherwise adjustments in the protection procedure need to be done. This idea basically the idea of the information-loss measure *lm* that was described in 4.1.

Also, it is interesting to evaluate the set of benchmarking indicators not only for the entire data set but also independently for subsets of the data. In this case the micro data are partitioned into a set of h groups. The evaluation of benchmarking indicators is then performed for each of the groups and the results are evaluated by looking at differences between indicators for original and modified data in each group. Templ et al. [2014a] gives for example a detailed description on the benchmarking indicators for the Structural Earnings Statistics data.

References

A. Alfons, S. Kraft, M. Templ, and P. Filzmoser. Simulation of close-to-reality population data for household surveys with application to EU-SILC. *Statistical Methods & Applications*, 20(3):383–407, 2011. URL http://dx.doi.org/10.1007/s10260-011-0163-2.

R. Brand. Microdata protection through noise addition. In *Privacy in Statistical Databases. Lecture Notes in Computer Science. Springer*, pages 347–359, 2004.

J. Burridge. Information preserving statistical obfuscation. *Statistics and Computing*, 13:321–327, 2003.

J. Domingo-Ferrer and V. Torra. A quantitative comparison of disclosure control methods for microdata. In *Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies*, pages 111–134, 2001.

M. Elliot. DIS: A new approach to the measurement of statistical disclosure risk. *Risk Management*, 2(4):39–48, 2000.

J. Gouweleeuw, P. Kooiman, L. Willenborg, and P-P. De Wolf. Post randomisation for statistical disclosure control: Theory and implementation. *Journal of Official Statistics*, 14(4):463–478, 1998.

A. Hundepool, A. Van de Wetering, R. Ramaswamy, L. Franconi, S. Polettini, A. Capobianchi, P-P. de Wolf, J. Domingo, V. Torra, R. Brand, and S. Giessing. *μ-Argus. User Manual*, 2008. Version 4.2.

D. Ichim and L. Franconi. Strategies to achieve sdc harmonisation at european level: Multiple countries, multiple files, multiple surveys. In *Privacy in Statistical Databases'10*, pages 284–296, 2010.

A. Kowarik, M. Templ, B. Meindl, and F. Fonteneau. *sdcMicroGUI: Graphical user interface for package sdcMicro.*, 2013. URL http://CRAN.R-project.org/package=sdcMicroGUI. R package version 1.0.3.

A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1), March 2007. ISSN 1556-4681. doi: 10.1145/1217299.1217302. URL http://doi.acm.org/10.1145/1217299.1217302.

A. Manning, D. Haglin, and J. Keane. A recursive search algorithm for statistical disclosure assessment. *Data Mining and Knowledge Discovery*, 16:165–196, 2008. ISSN 1384-5810. URL http://dx.doi.org/10.1007/s10618-007-0078-6. 10.1007/s10618-007-0078-6.

R. Maronna, D. Martin, and V. Yohai. *Robust Statistics: Theory and methods.* Wiley, New York, 2006.

J.M. Mateo-Sanz, F. Sebe, and J. Domingo-Ferrer. Outlier protection in continuous microdata masking. *Lecture Notes in Computer Science, Vol. Privacy in Statistical Databases, Springer Verlag*, 3050:201–215, 2004.

K. Muralidhar and R. Sarathy. Data shuffling- a new masking approach for numerical data. *Management Science*, 52(2):658–670, 2006.

K. Muralidhar, R. Parsa, and R. Sarathy. A general additive data perturbation method for database security. *Management Science*, 45:1399–1415, 1999.

Y. Rinott and N. Shlomo. A generalized negative binomial smoothing model for sample disclosure risk estimation. In *Privacy in Statistical Databases. Lecture Notes in Computer Science. Springer*, pages 82–93, 2006.

D.B. Rubin. Discussion: Statistical disclosure limitation. *J Off Stat*, 9(2):461–468, 1993.

P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical Report SRI-CSL-98-04, SRI International, 1998.

CJ. Skinner and DJ. Holmes. Estimating the re-identification risk per record in microdata. *Journal of Official Statistics*, 14:361–372, 1998.

L. Sweeney. k-anonymity: a model for protecting privacy. *Int J Uncertain Fuzziness Knowl Syst*, 10(5):557–570, 2002.

M. Templ. Estimators and model predictions from the structural earnings survey for benchmarking statistical disclosure methods. Research Report CS-2011-4, Department of Statistics and Probability Theory, Vienna University of Technology, 2011a. URL http://www.statistik.tuwien.ac.at/forschung/CS/CS-2011-4complete.pdf.

M. Templ. Comparison of perturbation methods based on pre-defined quality indicators. In *Joint UNECE/Eurostat work session on statistical data confidentiality, Tarragona, Spain*, Tarragona, 2011b. invited paper.

M. Templ and B. Meindl. Robust statistics meets SDC: New disclosure risk measures for continuous microdata masking. *Privacy in Statistical Databases. Lecture Notes in Computer Science. Springer*, 5262:113–126, 2008a. ISBN 978-3-540-87470-6, DOI 10.1007/978-3-540-87471-3_10.

M. Templ and B. Meindl. Robustification of microdata masking methods and the comparison with existing methods. *Privacy in Statistical Databases. Lecture Notes in Computer Science. Springer*, 5262:177–189, 2008b. ISBN 978-3-540-87470-6, DOI 10.1007/978-3-540-87471-3_15.

M. Templ and B. Meindl. Practical applications in statistical disclosure control using R. In J. Nin and J. Herranz, editors, *Privacy and Anonymity in Information Management Systems*, Advanced Information and Knowledge Processing, pages 31–62. Springer London, 2010. ISBN 978-1-84996-238-4. URL http://dx.doi.org/10.1007/978-1-84996-238-4_3. 10.1007/978-1-84996-238-4_3.

M. Templ, A. Kowarik, and B. Meindl. *sdcMicro: Statistical Disclosure Control methods for the generation of public- and scientific-use files. Manual and Package.*, 2013. URL http://CRAN.R-project.org/package=sdcMicro. R package version 4.0.4.

M. Templ, A. Kovarik, and B. Meindl. sdcmicro case studies. Research Report CS-2014-1, Department of Statistics and Probability Theory. Vienna University of Technology, 2014a. to be published soon.

M. Templ, B. Meindl, and A. Kowarik. Gui tutorial. Research Report CS-2014-2, Department of Statistics and Probability Theory. Vienna University of Technology, 2014b. to be published soon.