Mag. Bernhard Meindl
DI Alexander Kowarik
Priv.-Doz. Dr. Matthias Templ
office@data-analysis.at

data-analysis OG

Introduction to Statistical Disclosure Control (SDC)

Authors:

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

Vienna, January 10, 2014

Acknowledgement: International Household Survey Network (IHSN)*

*Special thanks to Francois Fontenau for his support and Shuang (Yo-Yo) CHEN for English proofreading

This document provides an introduction to statistical disclosure control (SDC) and guidelines on how to apply SDC methods to microdata. Section 1 introduces basic concepts and presents a general workflow. Section 2 discusses methods of measuring disclosure risks for a given micro dataset and disclosure scenario. Section 3 presents some common anonymization methods. Section 4 introduces how to assess utility of a micro dataset after applying disclosure limitation methods.

1 Concepts

A microdata file is a dataset that holds information collected on individual units; examples of units include people, households or enterprises. For each unit, a set of variables is recorded and available in the dataset. This section discusses concepts related to disclosure and SDC methods, and provides a workflow that shows how to apply SDC methods to microdata.

1.1 What is disclosure?

In general, disclosure occurs when an intruder uses the released data to reveal previously unknown information about a respondent. There are three different types of disclosure:

Identity disclosure: In this case, the intruder associates an individual with a released data record that contains sensitive information. Identity disclosure is possible through direct identifiers, rare combinations of values in the key variables and exact knowledge of continuous key variable values in external databases. For the latter, extreme data values (e.g., extremely high turnover values for an enterprise) lead to high re-identification risks.

Attribute disclosure: In this case, the intruder is able to determine some characteristics of an individual based on information available in the released data. For example, if all people aged 56 to 60 who identify their race as black in region 12345 are unemployed, the intruder can determine the value of the variable *labor status*.

Inferential disclosure: In this case, the intruder, though with some uncertainty, can predict the value of some characteristics of an individual more accurately with the released data.

If linkage is successful based on a number of identifiers, the intruder will have access to all of the information related to a specific corresponding unit in the released data. This means that a subset of critical variables can be exploited to disclose everything about a unit in the dataset.

1.2 Categorization of Variables

In accordance with disclosure risks, variables can be classified into three groups, which are not necessarily disjunctive:

Direct Identifiers are variables that precisely identify statistical units. For example, social insurance numbers, names of companies or persons and addresses are direct identifiers.

Key variables are a set of variables that, when considered together, can be used to identify individual units. For example, it may be possible to identify individuals by using a combination of variables such as gender, age, region and occupation. Other examples of confidential key variables are income, health status, nationality or political preferences. Key variables are also called implicit identifiers or quasi-identifiers. When discussing SDC methods, it is preferable to distinguish between categorical and continuous key variables based on the scale of the corresponding variables.

Non-confidential variables are variables that are not direct identifiers or key variables.

For specific methods such as l-diversity, another group of sensitive variables is defined in Section 2.3).

1.3 Remarks on SDC Methods

In general, SDC methods borrow 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. Statistical modeling methods form the basis of perturbation algorithms, to simulate data and quantify risks and information loss. Linear programming is used to modify data but minimize the impact on data quality.

Problems and challenges arise from large datasets and the need for efficient algorithms and implementations. Another layer of complexity is produced by complex structures of hierarchical, multidimensional data sampled with complex survey designs. Missing values are a challenge, especially for computation time issues; structural zeros also have impact on the application of SDC methods. Furthermore, the compositional nature of many components should always be considered, but adds even more complexity.

SDC techniques can be divided into three broad topics:

- Measuring disclosure risk (see Section 2)
- Measuring disclosure risk (see Section 3)
- Comparing original and modified data (information loss) (see Section 4)

1.4 Risk Versus Data Utility and Information Loss

The goal of SDC is always to release a safe micro dataset with high data utility and a low risk of linking confidential information to individual respondents. Figure 1 shows the trade-off between disclosure risk and data utility. We applied two SDC methods with different parameters to the Structural Earnings Statistics (SES) data [see Templ et al., 2014a, for more on anonymization of this dataset].

For Method 1, the parameter varies between 10 (small perturbation) to 100 (perturbation is 10 times higher). When the parameter value is 100, the disclosure risk is low since the data are heavily perturbed, but the information loss is very high, which also corresponds to very low data utility. When only low perturbation is applied to a dataset, both risk and data utility are high. It is easy to see that data anonymized with Method 2 have considerably lower risk; therefore, this method is preferable. In addition, information loss increases only slightly if the parameter

value increases; therefore, Method 2 with parameter value of approximately 7 would be a good choice in this case.

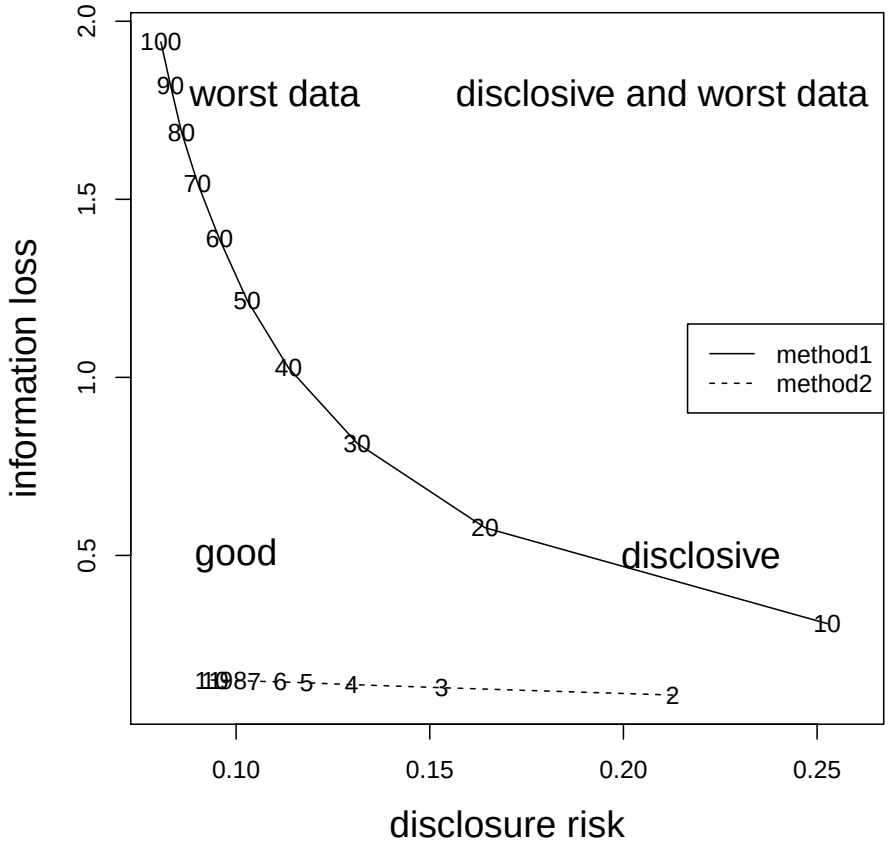

Figure 1: Risk versus information loss obtained for two specific perturbation methods and different parameter choices applied to SES data.

In real-world examples, things are often not as clear, so data anonymization specialists should base their decisions regarding risk and data utility on the following considerations:

What is the legal situation regarding data privacy? Laws on data privacy vary between countries; some have quite restrictive laws, some don't, and laws often differ for different kinds of data (e.g., business statistics, labor force statistics, social statistics, and medical data).

How sensitive is the data information and who has access to the anonymized data file? Usually, laws consider two kinds of data users: users from universities and other research organizations, and general users, i.e., the public. In the first case, special contracts are often made between data users and data producers. Usually these contracts explicitly forbid the usage of the data, except for very specific research projects, and allow data saving only within safe work environments. For these users, anonymized microdata files are called scientific use files, whereas data for the public are called public use files. Of course, the disclosure risk of a public use file needs to be very low, much lower than the corresponding risks 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 dataset. Data on individuals' medical treatments are more sensitive than an establishment's turnover values and number of employees. If the data contains very sensitive information, the microdata should have greater security than data that only contain information that is not likely to be attacked by intruders.

Which method is suitable for which purpose? While 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. Still other methods, such as swapping or post-randomization (PRAM), 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 datasets as well as the disclosure risks and data utility for anonymized versions of the data. To achieve good results (i.e., low disclosure risk, high data utility), it is necessary to anonymize in an explanatory manner by applying different methods using different parameter settings until a suitable trade-off between risk and data utility has been achieved.

1.5 Workflow

Figure 2 outlines the most common tasks, practices and steps required to obtain confidential data. The steps are summarized here:

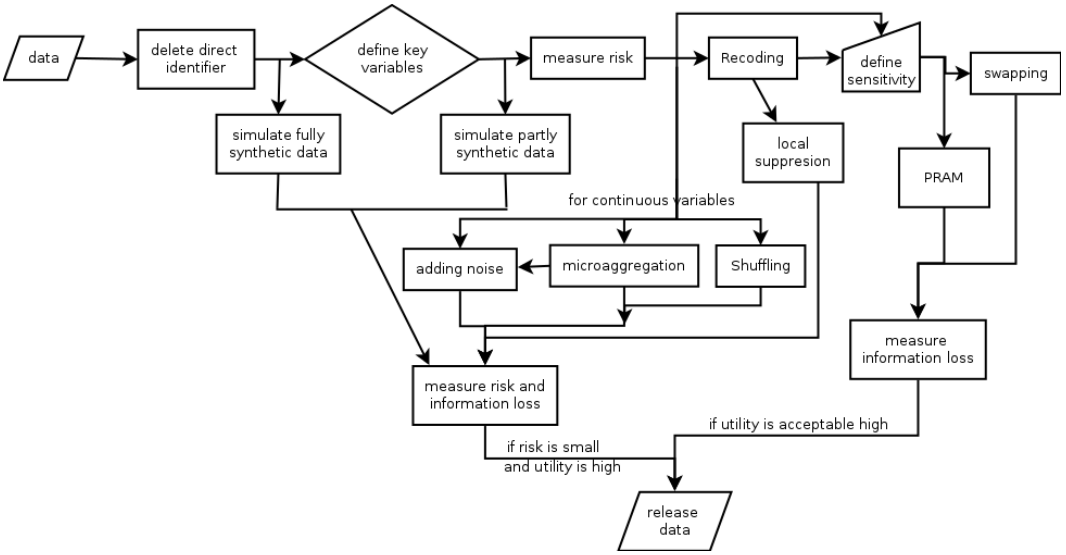

Figure 2: Possibilites for anonymising micro data using different SDC methods.

1. The first step is always to remove all direct identification variables and variables that contain direct information about units from the microdata set.

2. Second, determine the key variables to use for all risk calculations. This decision is subjective and often involves discussions with subject matter specialists and interpretation of 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, choosing key variables is not necessary

since all variables are produced synthetically, see for example Alfons et al. [2011].

3. After the selection of key variables, measure disclosure risks of individual units. This includes the analysis of sample frequency counts as well as the application of probability methods to estimate corresponding individual re-identification risks by taking population frequencies into account.

4. Next, modify observations with high individual risks. Techniques such as recoding and local suppression, recoding and swapping, or PRAM can be applied to categorical key variables. In principle, PRAM or swapping can also be applied without prior recoding of key variables; a lower swapping rate might be possible, however, if recoding is applied before. The decision as to 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 key variables is low. PRAM should be used if the number of key variables is large and the number of unique combinations is high; for details, see Sections 3.1 and 3.3 and for practical applications Templ et al. [2014a]. The values of continuously scaled key variables must be perturbed as well. In this case, micro-aggregation is always a good choice (see Section 3.4). More sophisticated methods such as shuffling (see Section 3.6) often provide promising results.

5. 5. After modifying categorical and numerical key variables of the microdata, estimate information loss and disclosure risk measures. The goal is to release a safe micro dataset with low risk of linking confidential information to individuals and high data utility. If the risks are low enough and the data utility is high, the anonymized dataset is ready for release. If not, the entire anonymization process must be repeated, either with additional perturbations if the remaining re-identification risks are too high, or with actions that will increase the data utility.

In general, the following recommendations hold:

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

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

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

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

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

Recommendation 6: Quantify the data utility not only by using typical estimates such as quantiles or correlations, but also by using the most important data-specific benchmarking indicators.

Recoding and micro-aggregation work 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 PRAM have been applied, these methods are advantageous whenever a large number of key variables is selected. This is because a high number of key variables leads to a high number of unique combinations that cannot be significantly reduced by applying recoding.

1.6 R-Package sdcMicro and sdcMicroGUI

SDC methods introduced in this guideline can be implemented by the **R**-Package `sdcMicro`. Users who are not familiar with the native **R** command line interface can use `sdcMicroGUI`, an easy-to-use and interactive application. For details, see Templ et al. [2014b, 2013].

2 Measuring the Disclosure Risk

Measuring risk in a micro dataset is a key task. Risk measurements are essential to determine if the dataset is secure enough to be released. To assess disclosure risk, one must make realistic assumptions about the information data users might have at hand to match against the micro dataset; these assumptions are called disclosure risk scenarios. Based on a specific disclosure risk scenario, it is necessary to define a set of key variables (i.e., identifying variables) that can be used as input for the risk evaluation procedure.

2.1 Population Frequencies and the Individual Risk Appoach

Typically, risk evaluation is based on the concept of uniqueness in the sample and/or in the population. The focus is on individual units that possess rare combinations of selected key variables. The assumption is that units having rare combinations of key variables can be more easily identified and thus have a higher risk of re-identification. It is possible to cross-tabulate all identifying variables and view their cast. Keys possessed by only very few individuals are considered risky, especially if these observations also have small sampling weights. This means that the expected number of individuals with these patterns is expected to be low in the population as well.

To assess whether a unit is at risk, a threshold approach is typically used. If the 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. Let us define frequency counts 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 (i.e., *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 unique or zero in the sample.

F_i is usually not known, since, in statistics, information on samples is collected to make inferences about populations.

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*. Because the values of observations 1 and 8 are equal and therefore the sample frequency counts are $f_1 = 2$ and $f_8 = 2$. 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 (key) $(1, 2, 5, 1)$ exist in the sample and 110 observations with this pattern (key) 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

One can show, however, that these estimates almost always overestimate small population frequency counts [see, e.g., Templ and Meindl, 2010]. A better approach is to use so-called super-population models, in which population frequency counts are modeled given certain distributions. For example, the estimation procedure of sample counts given the population counts can be modeled by assuming a negative binomial distribution [see Rinott and Shlomo, 2006] and is implemented in `sdcMicro` in function measure_risk() [see Templ and Meindl, 2010]. This calculation can also be done from within the graphical user interface.

2.2 The Concept of k-anonymity

Based on a set of key variables, one desired characteristic of a protected micro dataset 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 microdata. This is equal to $f_i \geq k\ , i = 1, ..., n$. A typical value is $k = 3$.

k-anonymity is typically achieved by recoding categorical key variables into fewer categories and by suppressing specific values of key variables for some units; see

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

Section 3.1 and 3.2.

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/keys in the key variables and let the group fulfill k-anonymity. A data intruder can therefore by definition not identify an individual within this group. If all observations have the same entries in an additional sensitive variable, however (e.g., cancer in the variable medical diagnosis), an attack will be successful if the attacker can identify at least one individual of the group, as the attacker knows that this individual has cancer with certainty. The distribution of the target-sensitive variable is referred to as l-diversity.

Table 2 considers a small example dataset that highlights the calculations of 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 appear in the fourth column. They equal 3 for the first three observations; the fourth observation is unique and frequency counts f_i are 2 for the last two observations. Only the fourth observation 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 the intruder still knows the exact information of the sensitive variable. For these observations, the l-diversity measure is 1, indicating that sensitive information can be disclosed, since the value of the sensitive variable is $= 62$ for both of these observations

Differences in values of sensitive variables can be measured differently. We present here the distinct diversity that counts how many different values exist within a pattern. Additional methods such as entropy, recursive and multi-recursive are implemented in the software. For more information, see the help files of `sdcMicro`.

2.4 Sample Frequencies on Subsets: SUDA

The Special Uniques Detection Algorithm (SUDA) estimates disclosure risks for each unit. SUDA2 [e.g., Manning et al., 2008] is a recursive algorithm to find Minimal Sample Uniques (MSUs). SUDA2 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 primarily depends on two aspects:

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

(b) (b) The larger the number of minimal sample uniqueness 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*, which is the maximum size of variable subsets of the key variables, $MSUmin_i$ is the number of MSUs of observation and i and n are the number of observations of the dataset. 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 observations contained in this pattern; this summation is the contribution of (b).

The final SUDA score is calculated by normalizing these SUDA scores by dividing them by $p!$, with p being the number of key variables. To receive the so-called Data Intrusion Simulation (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. This algorithm calculates the probabilities of correct matches given unique matches. It is, however, out of scope to precisely describe this algorithm here; reference Elliot [2000] for details. The DIS SUDA score is calculated from the SUDA and DIS scores, and is available in `sdcMicro` as `disScore`).

Note that this method does not consider population frequencies in general, but does consider sample frequencies on subsets. The DIS SUDA scores approximate uniqueness by simulation based on the sample information population, but to our knowledge, they generally 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 dataset as in Section 2.1. Sample frequency counts f_i as well as the SUDA and DIS SUDA scores have been calculated. The SUDA scores have the largest value for observation 4 since subsets of key variables of this observation are also unique, while for observations $1-3$, $5-6$ and 8, no subsets are unique.

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 an output of this function. The contribution to the SUDA score is calculated by assessing how often a category of a key variable contributes to the score.

In Table 4, all concepts previously discussed have been applied. The table lists estimations of frequency counts for the sample and population, which corresponds to the sum of sampling weights for each group, the l-diversity measure, the SUDA algorithm and the individual risk estimation. Note 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

2.5 Calculating Cluster (Household) Risks

Micro datasets often contain hierarchical cluster structures; an example is social surveys, when individuals are clustered in households. The risk of re-identifying an individual within a household may also affect the probability of disclosure of other members in the same household. Thus, the household or cluster-structure of the data must be taken into account when calculating risks

It is commonly assumed that the risk of re-identification 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`.

2.6 Measuring the Global Risk

Sections 2.1 through 2.5 discuss the theory of individual risks and the extension of this approach to clusters such as households. In many applications, however, estimating a measure of global risk is preferred. Any global risk measure will result in one single number that can be used to assess the risk of an entire micro dataset.

2.6.1 Measuring the global risk using individual risks

Two approaches can be used to determine the global risk for a dataset using individual risks:

Benchmark: This approach counts the number of observations that 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 · median absolute deviation (MAD) of all unit risks.

Global risk: The sum of the individual risks in the dataset gives the expected number of re-identifications [see Hundepool et al., 2008].

The benchmark approach indicates whether the distribution of individual risk occurrences contains extreme values; it is a 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 whether some unit risks behave differently compared to most of the other individual risks. The global risk approach is based on an absolute measure of risk. Following is the print output of the corresponding function from `sdcMicro`, which shows both measures:

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

If a cluster (e.g., a household ID) has been defined, a global risk measurement taking into account this hierarchical structure is also reported.

2.6.2 Measuring the global 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, global risk can be defined as the following [see also 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 primary effects and possible interactions. This 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 using, for example, iterative proportional fitting. The quality of this risk measurement approach depends on the number of different keys that result from cross-tabulating 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 must also be considered that if the model for prediction is weak, the quality of the prediction of the frequency counts is also weak. Thus, the risk measurement with log-linear models may lead to acceptable estimates of global risk only if not too many key variables are selected and if good predictors are available in the dataset.

In `sdcMicro`, global risk measurement using log-linear models can be completed with function LLmodGlobalRisk(). This function is experimental and needs further testing, however. It should be used only by expert users.

2.7 Measuring Risk for Continuous Key Variables

The concepts of uniqueness and k-anonymity cannot be directly applied to continuous key variables because almost every unit in the dataset will be identified as unique. As a result, this approach will fail. The following sections present methods to measure risk for continuous key variables.

2.7.1 Distance-based record linkage

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 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 microdata with other external data sources.

We assume that an intruder has information about a statistical unit included in the microdata; the intruder's information overlaps on some variables with the information in the data. In simpler terms, we assume that the intruder's information can be merged with microdata that should be secured. In addition, we also assume that the intruder is sure that the link to the data is correct, except for micro-aggregated data (see Section 3.4). 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 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 microdata with other external data sources.

We assume that an intruder has information about a statistical unit included in the microdata; the intruder's information overlaps on some variables with the information in the data. In simpler terms, we assume that the intruder's information can be merged with microdata that should be secured. In addition, we also assume that the intruder is sure that the link to the data is correct, except for micro-aggregated data. 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 centered on the masked value. Then they calculate the length of the intervals based on the standard deviation of the variable under consideration (see Figure 3, upper left graphic).

2.7.2 Special treatment of outliers when calculating disclosure risks

Almost all datasets used in official statistics contain units whose values in at least one variable are quite different from the general observations. As a result, these variables are very asymmetrically distributed. Examples of such outliers might be enterprises with a very high value for turnover or persons with extremely high income. In addition, multivariate outliers exist [see Templ and Meindl, 2008].

Unfortunately, intruders may want to disclose a large enterprise or an enterprise with specific characteristics. Since enterprises are often sampled with certainty or have a sampling weight close to 1, intruders can often be very confident that the enterprise they want to disclose has been sampled. In contrast, an intruder may not be as interested to disclose statistical units that exhibit the same behavior as most other observations. For these reasons, it is good practice to define measures of disclosure risk that take the outlyingness of an observation into account. For details, see Templ and Meindl [2008]. 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 (called RMDID2 in Figure 3) is also included in the `sdcMicro` package. It works as described in Templ and Meindl [2008] and is listed as follows:

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

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

3. 3. Check whether the corresponding masked values of a unit fall into the intervals around the original values. If the masked value lies within such an interval, the entire observation is considered unsafe. We obtain a vector indicating which observations are safe or which are not. For all unsafe units, at least m other observations from the masked data should be very close. Close is quantified 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 depicts the idea of weighting disclosure risk intervals. For simple methods (top left and right graphics), the rectangular regions around each value are the same size for each observation. Our proposed methods take the *RMD*s of each observation into account. The difference between the bottom right and left graphics 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, check whether this observation has close neighbors. 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 *safe*.

These methods are also implemented and available in sdcMicro as **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, there are two kinds of anonymization methods: deterministic and probabilistic. For categorical variables, recoding and local suppression are deterministic procedures, while swapping and PRAM [Gouweleeuw et al., 1998] are based on randomness and considered probabilistic methods. For continuous variables,

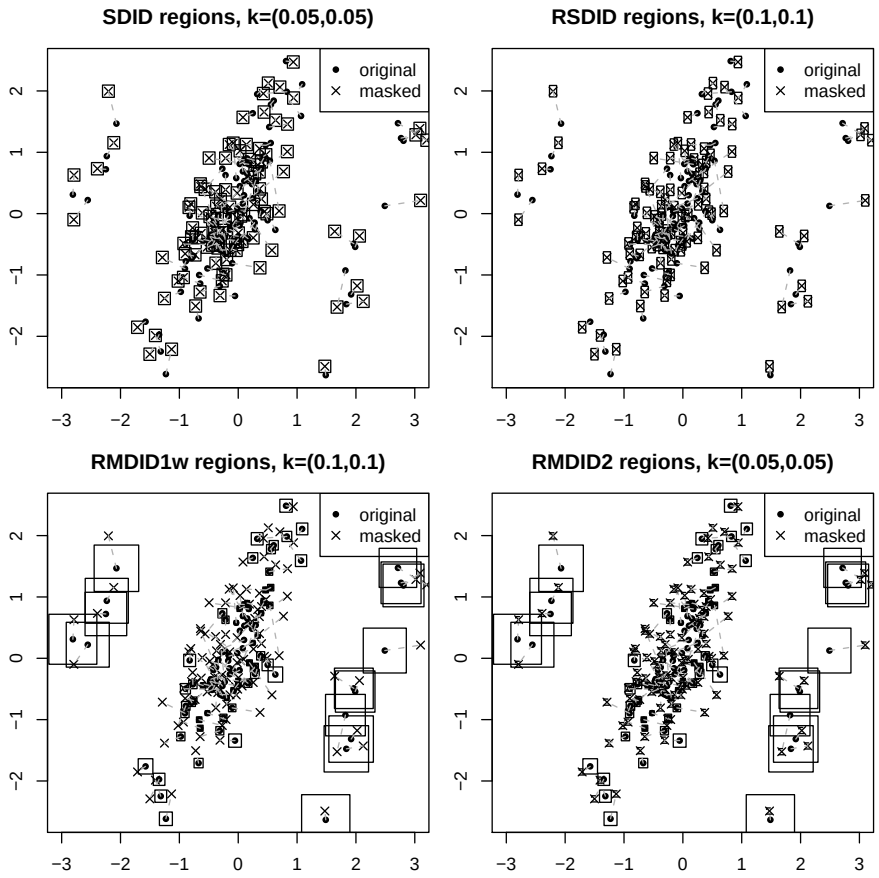

Figure 3: Original and corresponding masked observations (perturbed by adding additive noise). In the bottom right graphic, small additional regions are plotted around the masked values for *RMDID2* procedures.

micro-aggregation 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 software's pseudo random number generator should be fixed to ensure reproducibility of the results.

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 goal 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 (i.e., 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 idea for this approach is that all values above (i.e., top coding) and/or below (i.e., bottom coding) a pre-specified threshold value are combined into a new category. Function **globalRecode**() can be applied in `sdcMicro` to perform both global recoding and top/bottom coding. A help file with examples is accessible using ?**globalRecode**. Note that `sdcMicroGUI` offers a more user-friendly way of applying global recoding.

3.2 Local Suppression

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

Using function **localSupp**() of `sdcMicro`, it is possible to suppress the values of a key variable for all units having individual risks above a pre-defined threshold, given a disclosure scenario. 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, one can use function **localSuppression**(). This algorithm also allows specification of a user-dependent reference that determines which key variables are preferred when choosing values that need to be suppressed.

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].

By specifying the importance of variables as a parameter in function **localSuppression**(), for key variables with high importance, suppression will only take place if no other choices are possible. This is useful if, for example, a scientific use file with specific requirements must be produced. Still, it is possible to achieve k-anonymity for selected key variables in almost all cases.

3.3 Post-randomization (PRAM)

Post-randomization [Gouweleeuw et al., 1998] PRAM is a perturbation, probabilistic method that can be applied to categorical variables. The 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. For each category of a categorical variable, this matrix lists probabilities to change into other possible categories.

As an example, consider 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 $p_1 = 0.15$ can a value of A1 be changed to either A2 or A3. The probability of a change from category A1 to A2 might be fixed with probability $p_2 = 0.1$ and changes from A1 to A3 with $p_3 = 0.05$. Probabilities to change values from class A2 to other classes and for A3, respectively, must be specified beforehand. All transition probabilities must be stored in a matrix that is the main input to function **pram**() in sdcMicro. This following example uses the default parameters of **pram**() rather than a custom transition matrix. 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
Number of changed observations:
- - - - - - - - - - -
x != x_pram : 1 (6.67%)
```

This summary provides more details. It shows a table of original frequencies as well as the corresponding table after applying the PRAM procedure. All transitions that took place are also 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         5
2          2         5
3          3         5
```

PRAM is applied to each observation independently and randomly. 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 PRAM 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.

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

3.4 Microaggregation

Micro-aggregation is a perturbative method that is typically applied to continuous variables. The 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, but other robust methods are also possible. Individual values of the records for each variable are replaced by the group aggregation value, which is often the mean; as an example, see Table 5, where two values that are most similar are replaced by their column-wise means.

Table 5: Example of micro-aggregation. Columns 1-3 contain the original variables, columns 4-6 the micro-aggregated 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

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

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

3.5 Adding Noise

Adding noise is a perturbative protection method for microdata, which is typically applied to continuous variables. This approach protects data against exact matching with external files if, for example, information on specific variables is available from registers.

While this approach sounds simple in principle, many 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 usually 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 respected, however, 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 co-variance matrix of the original microdata. In the case of correlated noise addition, correlation coefficients are preserved and at least the co-variance matrix can be consistently estimated from the perturbed data. The data structure may differ a great deal,

however, if the assumption of normality is violated. Since this is virtually always the case when working with real-world datasets, a robust version of the correlated noise method is included in `sdcMicro`. This method allows departures from model assumptions and is described in detail in citeTempl08f). More information can be found in the help file by calling ?**addNoise** or using the graphical user interface help menu.

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 ensure that the amount of noise added takes into account the underlying sample size and sampling weights. Noise can be added to variables in `sdcMicro` using function lstinlineaddNoise() or by using `sdcMicroGUI`.

3.6 Shuffling

Various masking techniques 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.

Shuffling [Muralidhar and Sarathy, 2006] simulates a synthetic value 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). In the implementation of sdcMicro, a model of almost any form and complexity can be specified (see ?**shuffling** for details).

4 Measuring Data Utility

Measuring data utility of the microdata set after disclosure limitation methods have been applied is encouraged 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, use various classical estimates such as means and covariances. 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 perturbed data and then compared. Following are three important information loss measures:

- **IL1s** is a measures introduced by [Mateo-Sanz et al., 2004]. 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 co-variances from standardized continuous key variables of the original and perturbed variables. Eigenvalues can be estimated from a robust or classical version of the co-variance 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 *sdcMicroObj* is generated or whenever continuous key variables are modified 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 same structure as the original file. An important goal, however, should always be that the difference in results of the most important statistics based on anonymized and original data should be very small or even zero. Thus, the goal 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 evaluate what users of the underlying data are analyzing and then try to determine the most important estimates, or *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 dataset. Indicators that refer to the most sensitive variables within the microdata should also 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 as to how to compare the indicators
- Calculation of all benchmarking indicators of 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 as to whether the data utility of the protected micro dataset 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 dataset 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 applied procedures or start from scratch and completely change the anonymization process.

Usually the evaluation is focused on the properties of numeric variables, given unmodified and modified microdata. It is of course also possible to review 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 idea is to predict important, sensitive variables using this model both for the original and protected micro dataset as 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 microdata, then the resulting quality is assessed. If the deviations are small enough, one may go on to publish the safe and protected micro dataset. Otherwise, adjustments must be made in the protection procedure. This idea is similar to the information loss measure **lm** described in Section 4.1.

In addition, it is interesting to evaluate the set of benchmarking indicators not only for the entire dataset but also independently for subsets of the data. In this case, the microdata 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 reviewing differences between indicators for original and modified data in each group. Templ et al. [2014a] gives a detailed description of benchmarking indicators for the SES 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, 2008. ISBN 978-3-540-87470-6, DOI 10.1007/978-3-540-87471-3_10.

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 4.1.1.

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.