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

Tutorial for `sdcMicroGUI` (and `sdcMicro`)

Matthias Templ, Bernhard Meindl and Alexander Kowarik

Vienna, December 6, 2016

http://www.data-analysis.at

Acknowledgement: International Household Survey Network (IHSN)*

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

1 Overview of sdcMicroGUI

The sdcMicroGUI [Kowarik et al., 2013] serves as an easy-to-handle, highly interactive tool for users who want to use the sdcMicro package for statistical disclosure control but are not familiar with the native R command line interface. The software performs automated recalculation and display of frequency counts, individual and global risk measures, information loss and data utility after each anonymization step. Changes to risk and utility measurements of the original data are also conveniently displayed in the graphical user interface (GUI). Furthermore, the code of every anonymization step carried out within the GUI is saved in a script, which can easily be exported, modified and re-used, making it possible to reproduce any results.

The sdcMicroGUI package has the following capabilities:

Link to sdcMicro: The GUI uses the functionality of the sdcMicro [Templ et al., 2013b] package. It allows high performance and fast computations, since all basic operations are written in either C or C++.

Import/Export: Datasets exported from other statistical software, such as SAS, SPSS, Stata, can easily be imported into sdcMicroGUI. It is also possible to use .csv files as well as data stored in R binary format. An interactive preview and selection of import parameters (such as delimiters or separators) are provided for the import of .csv files, which allows users to read the data correctly into the GUI. Export facilities are provided for the same formats from which data can be read into the GUI.

Usability: The package is an easy-to-use tool for anonymization of microdata, and all methods are easily accessible.

Recoding: Facilities to rename and regroup categories and change values of a variable are included.

Interactivity: Risk und utility measures are automatically estimated and displayed whenever users apply a disclosure limitation technique. In this way, users can immediately see the effects of any action. In addition, the risk and utility of the original unmodified data are displayed, which helps the user assess the effectiveness of the anonymization.

Undo Button: Because users can undo the last step completed in the GUI, they can try out several methods with different parameters and get instant feedback until the best result is achieved. Currently, users can reverse exactly one step in the history.

Reporting: Automatically generated, standardized reports in various output formats can be produced directly from the user interface. Reports can be exported to html, LaTex or plain text files. Users can generate two different types of reports. The more detailed, internal version includes the anonymization methods applied as well as estimates of risk and utility; there are different outputs depending on the methods used for different methods. The shorter, external version provides a brief summary of the anonymization procedure applied, suitable for external viewers. In this type of report, detailed comparisons and summaries are not included. For more information, see Section 8.2.

Reproducibility: With `sdcMicroGUI`, users can save, load or edit scripts for later re-use. Within the GUI, each step of the anonymization procedure is recorded and stored in a script. The script includes valid `R`-expressions that can be copied into `R`. Thus, any anonymization procedure can be reproduced either by loading a script into the GUI or pasting the script directly into an `R`-console.

2 Installation and Updates

The recommended procedure to install the software consists of the following steps:

Install R: If R is already installed on the computer, ensure that it is the current version. If the software is not installed, go to http://cran.r-project.org/bin/ and choose your platform. For Windows, just download the executable file and follow the on-screen instructions.

Install sdcMicro and sdcMicroGUI: Open R on your computer and type:

```
install.packages("sdcMicroGUI")
```

Installation is needed only once. Note that the GUI requires the `GTK+` package to draw windows. When installing sdcMicroGUI, all required packages (including `GTK+`) are automatically installed if the user has sufficient system administration rights.

Update: Typing **update.packages**() into R searches for possible updates and installs new versions of packages if any are available. Users can also click on the menu item *GUI* → *Check for Updates*; this should be done regularly.

This will allow you to install the packages.

3 Open sdcMicroGUI and Import Data

3.1 Open sdcMicroGUI

Open the software R and type:

```
require(sdcMicroGUI); sdcGUI()
```

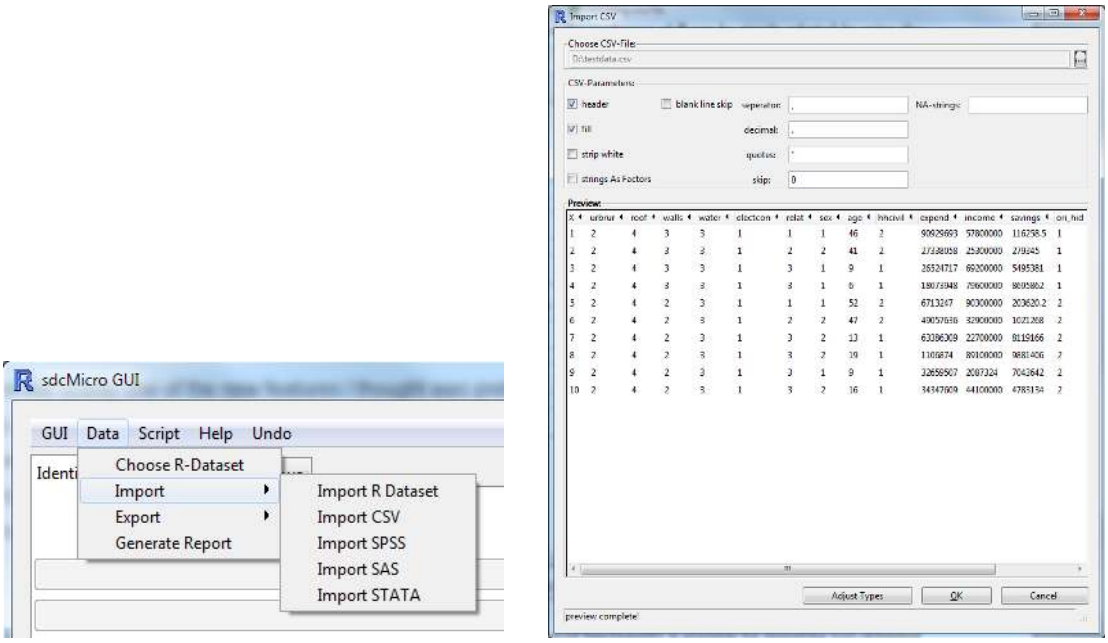
This will load the `sdcMicroGUI` package into R and display the point-and-click GUI. If you have not installed sdcMicroGUI, you will see an error message; follow the steps described in Section 2 to install the package.

3.2 Key Elements of the GUI

Main menu

Figure 1 shows the top menu of `sdcMicroGUI`. Following are the elements of the main menu, which is located at the top of the GUI and always visible:

- **GUI** – This menu item closes or restarts the interface and checks for updates of `sdcMicro` and/or `sdcMicroGUI`.



(a) The *data*-menu entry at the main menu.

(b) On-the-fly preview of `.csv` files.

Figure 1: *Data* menu entry and the on-the-fly preview when importing .csv-files.

- **Data** – This is used to import data into the GUI from various data formats as well as existing R-objects. Figure 1(b) shows the import mask for .csv files. One can also use this menu to export data to different data formats and generate a report.
- **Script** – This menu saves, loads or views a script generated by `sdcMicroGUI`.
- **Help** – This menu provides different resources, such as information on disclosure control methods or documentation on the underlying functions of `sdcMicro`.
- **Undo** – The button *Undo* allows users to go back one step in the anonymization process. This makes it is possible to try out an anonymization method; if the results are not satisfying, the last action can be reversed easily.

GUI Main Window

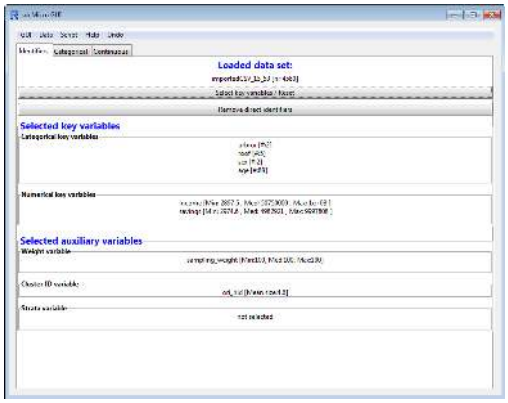
The GUI displays three tabs (see Figures 2(a), 2(b) and 2(c)).

Identifiers – The first tab summarizes the current selection of key variables after a dataset is imported. It displays the number of categories for each categorical key variable, and three statistics (i.e., minimum, median and maximum) for each numerical key variable. If any auxiliary variables, such as sampling weight, household ID or a strata variable, are selected, summary statistics are shown for these variables as well. In this tab, direct identifiers can be removed by clicking on the corresponding button. It is also possible to reset the current choice of key variables (see Figure 2(a)).

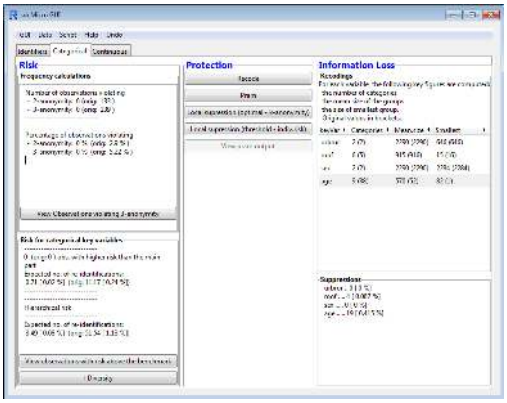
Categorical – This tab is divided into three parts. Information about disclosure risk based on frequency counts is shown at the top left of Figure 2(b). Be-

neath this information display additional risk measures, including the number of observations having a much greater risk than the primary observations (using robust measures), as well as the expected number of re-identifications (i.e., the sum of all individual risks) for both the original and the modified microdata. The expected number of re-identifications based on a hierarchical risk measure exploiting household information is also calculated and displayed. Methods that can be used for the anonymization of categorical key variables (e.g., recoding, post-randomization [PRAM], local suppression) are displayed in the center of the screen. Important measures on information loss are shown on the right. Information about recoding (e.g., number of categories, mean size and number of observations in the smallest category) for both original and modified key variables is listed at the top right. The number of suppressions within each key variable appears at the bottom right.

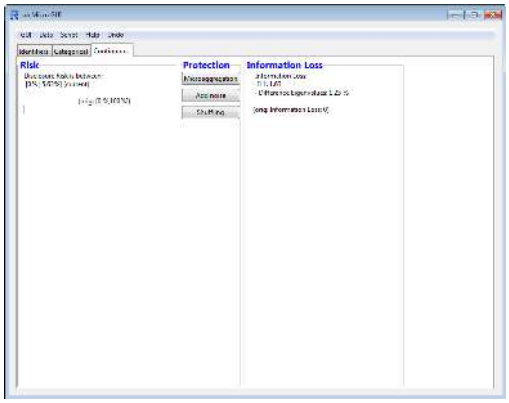
Continuous: The third tab is also divided into three parts (see Figure 2(c)). A risk measure that calculates the percentage of masked data points that are too close to the original data values is shown on the left. Methods such as microaggregation, adding noise and shuffling, which can be used to anonymize continuous key variables, are shown in the center and can be selected and applied. On the right, two measures of information loss, IL1 and differences in eigenvalues, are listed.



(a) View of Tab "Identifiers"



(b) View on Tab "Categorical"



(c) View on Tab "Continuous"

Figure 2: The three main tabs of the GUI. See Section 3.2 for information on content.

All GUI frames and views that present summaries, names, frequency calculations, suppressions, disclosure risk and data utility are filled in with actual values as soon as data are selected. Moreover, buttons to apply certain methods such as *recoding*, *PRAM* or *local suppression* become clickable when data are loaded into the GUI. As soon as a method is applied on the data, all related views and measures are updated with current values. For example, after applying global recoding, the disclosure risk and data utility for categorical key variables are updated to show current values automatically.

3.3 Select, Load or Import Data

The GUI offers several options for importing data into the system; see Figure 1(a). Data that is already available in the `R` workspace can simply be selected by using the menu entry *Data* → *Choose R-Dataset*. Using *Data* → *Import*, it is possible to import data in various formats, such as native RData files, as well as import/export files from other statistical software products, such as `SPSS`, `SAS` and `STATA`.

The advanced functionality available to import text-delimited *.csv* files is very important (see Figure 1(b)). In this case, the user is presented with a data preview window showing the first rows of a dataset with the current data import parameters and several ways to change import options, such as:

- *header*: If checked, the first line of a dataset displays column names
- *fill*: If checked, blanks are added for rows of unequal length
- *strip white*: Allows leading and trailing white space to be stripped from unquoted character fields
- *strings as factors*: If checked, character vectors are converted to factors
- *blank line skip*: If checked, blank lines are ignored when reading the file

Additionally, the separator between values, decimal operator, quotes, skip and coding of missing values (i.e., *NA*-strings) can be specified. When the user changes a field, the preview window of the data changes according to the options and informs the user whether the data was correctly imported. The type of each variable can also be specified (i.e., *numeric* or *factor*[1]) when importing a *.csv* file, using the button *Adjust Types*, as shown at the bottom of Figure 1(b).

4 Removing Direct Identifiers

After a dataset is selected or imported, the buttons *Select key variables/Reset* and *Remove direct identifiers* under the GUI's "Identifiers" tab become active (see Figure 2(a)). Click on the button *Remove direct identifiers* to select the variables that specifically identify statistical units and remove them.

[1] In `R`, data type *numeric* belongs to continuous variables while data type *factor* belongs to categorical variables with given levels

5 Selecting Key Variables

Once a dataset is selected or imported, the "Select variables" window automatically pops up (see Figure 3). To illustrate the selection of key variables, the `sdcMicro` test data set is chosen. This is a real survey used for the development of `C++` code at the International Household Survey Network.

Click on *Select key variables/Reset* under the "Identifiers" tab to modify your selection. Note that statistical methods and the corresponding functions in `R` are in most cases specific in terms of the scale of variables. Some methods on SDC should be applied only to categorical variables (in `R`, these variables correspond to vectors of class *factor*), while some are suitable only for continuous variables (in `R`, they are vectors of type *numeric*). In any case, the key variables should be selected.

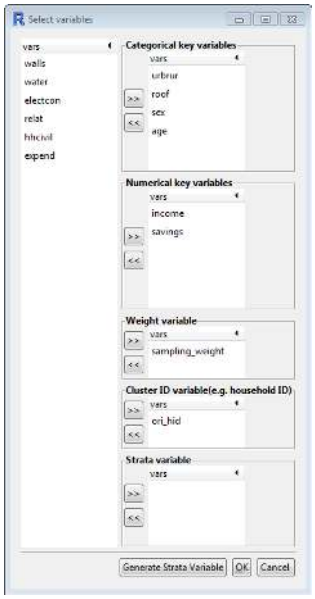


Figure 3: Select variables window.

For our chosen data set, the variables *urbr* (degree of urbanisation), *roof* (household has roof), *sex* and *age* (in years) have been chosen as categorical key variables (a discussion on the selection of key variables is given in the Introduction to Statistical Disclosure Control (REFERENCE)). As continuous key variables the variables and *income* and *savings* are chosen, i.e. a scenario where we believe that information on this variables can be stored in other available data bases and therefore be used for disclosing information.

Information on **clustering** (e.g., households) is often required and can be entered by using the variable selection window. In the chosen data set this belongs to variable *ori_hid*, which is an ID that express which people lives in the same household. It may also be important to select a **weight vector**, especially if the microdata have been collected from a complex survey. This information must be provided by the user in order for the system to make use of this knowledge. For our test data set this information is stored in variable *sampling_weights*.

Some functions can also be applied on strata (on domain level). In this case, specifying or creating a variable defining population subgroups is required.

It is often necessary to generate a new stratification variable that combines a few categorical variables. This can be done by clicking the button *Generate Strata Variable*, see Figure 3. A window pops up where users can specify the variables that should be used for the stratification of the data (see Figure 4).

Other variables do not contribute to any aspect in the anonymization process[2].

Note that most of the variable selections are optional. For example, users are not required to select any continuous (or *numerical*, in the GUI) variables if they are not present in the data.

[2]Except for when applying the method of shuffling, all variables are made selectable (as predictors), independent of this choice

If variables have the wrong scale (e.g., if categorical key variables are saved as *numeric*), the global recoding frame automatically pops up and can then be used to recode the corresponding variables.

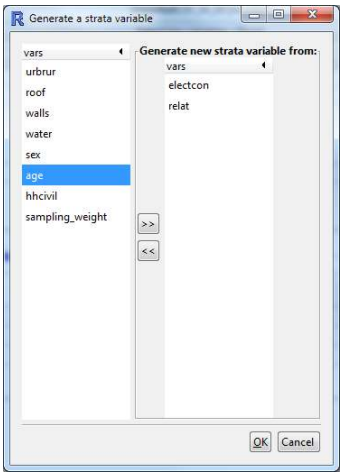


Figure 4: Generate a strata variable window.

6 Anonymisation of Categorical Key Variables

Figure 2(a) shows the GUI after the necessary variables have been selected and optionally recoded. The *Categorical* tab, which appears second on the `sdcMicroGUI` main window, shows options for categorical key variables (see Figure 2(b)).

The left column of Figure 2(b) shows frequency counts and risks for categorical key variables. This information is updated whenever a method is applied to categorical key variables to show how recoding impacts frequency counts and individual risks. All observations violating 3-anonymity [Samarati, 2001, Sweeney, 2002] can be shown by clicking on the button *View Observations violating 3-anonymity*. To compare the impact of anonymization methods already applied, the same information calculated with the original, unmodified data is also displayed. This can also be seen in Figure 2(b) as for example 133 (2.9%) of the observations have been violating 2-anonymity and 239 (5.22%) observations have been violating 3-anonymity.

Information on the number of observations expected to be re-identified under a given selection of key variables is displayed (as the sum over all individual disclosure risks), as well as information on the number of observations with considerably higher risk than the main part of all observations. Observations with high individual risks can be viewed by clicking the button *View observations with risk above the benchmark*. The l-diversity measure [Machanavajjhala et al., 2007] can also be calculated by clicking on the corresponding button; a new window pops up in which the user can select sensitive variables (for example income or savings in this case) and set the l-recursive constant (see Figure 6(a)). When the button *OK* in this window is clicked, another window containing the results pops up and the distribution of l-diversity scores for each sensitive variable is displayed. By clicking on the button *View Observations violating* 2*-diversity*, another window is opened that includes a table showing all observations that are violating 2-diversity, which means that only up to two different values of the sensitive variables exist in specific keys. If no observations violate 2-diversity, an information message is shown. We note that all these measures are explained in more detail in the corresponding disclosure guidelines that are also included with **sdcMicro**.

The middle column of Figure 2(b) shows four statistical disclosure control (SDC)

methods for anonymization of categorical key variables: *recoding*, *PRAM* [Gouweleeuw et al., 1998] and two different methods to perform *local suppression*, which will be explained in the following sections.

Information about the effects of recoding in key variables is displayed on the right side of Figure 2(b). For each key variable, the number of categories, mean sizes and size of the smallest category are shown. The same values based on the original, unmodified variables are also displayed, in parentheses. The number and percentages of suppressions for each categorical key variable appear below this information. Figure 2(b) shows, for example, that the original number of categories of key variable *age* was 88, the number of categories after recoding was reduced to 9. The mean size of age-categories after recoding is 570 while it was only 52 before recoding. Similarily, the size of the smallest age category after recoding is 82 while it was 1 before recoding.

6.1 Recoding

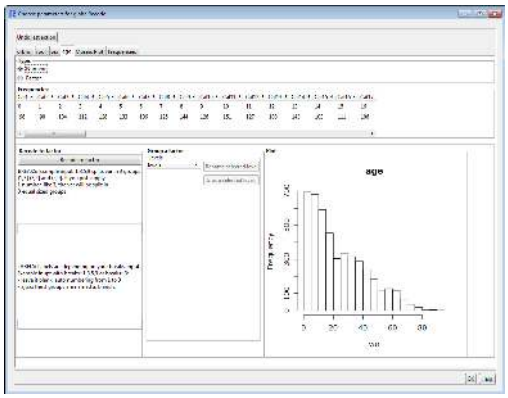
Clicking on the button *Recode* in the *Categorical* tab opens a window in which the categorical key variables can be recoded, as shown in Figure 5.

It is possible to recode all categorical key variables separately. Corresponding variable names are visible in the menu of the key variables configuration frame. Any variable can now be converted, recoded, grouped and renamed. Breaks and label names can be specified for converting continuous scaled variables into classes. It is also possible to group categories of factors into broader categories by using the button *Group selected level*, and to rename specific categories by clicking on the button *Rename selected level* after selecting a factor level. To demonstrate this, we explain how to recode variable *age* from the testdata set that is included with **sdcMicro** in more detail. The variable holds the age of respondents in years. To form for example an age group holding all the respondents between 20 and 29 years one would have to select the corresponding years. Once the desired levels have been selected the button *Group selected level* must be clicked. A window pops up in which one can enter a desired level-name for this group.

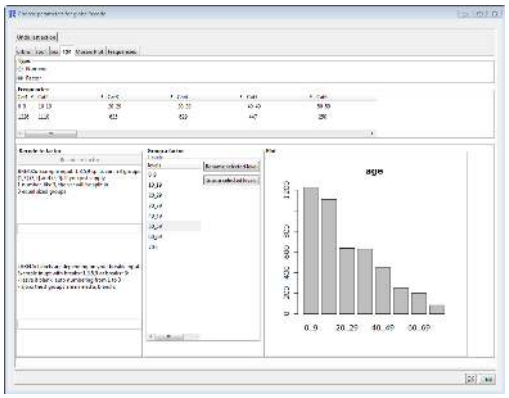
Distribution of the variables and information on tabulated variable are always shown graphically. Additionally, the frequency counts of all key variables are available and visible in a separate tab.

Sample and population frequencies (f_k and F_k, respectively) are illustrated when the *Frequencies* tab in the global recoding menu is clicked. Frequencies, individual risks [Franconi and Polettini, 2004] and values of the categorical key variables are shown. The table containing these statistics is interactive and sortable; therefore, for example, clicking on the top of the risk column sorts the table according to the values of the individual risks in ascending order. Clicking a second time will sort the table by this variable in descending order. This is especially helpful because on can sort the table according to individual reidentification risks of observations. Thus, it is very easy to identify the observations having the highest re-identification risks.

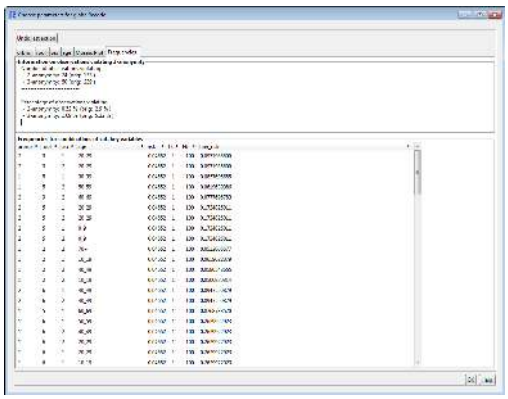
In tab $\longrightarrow$ *Mosaic Plot* of the global recoding menu, a mosaic plot of all selected key variables is shown. The plot (which is definately for experts) shows the multivariate distribution of the selected categorical key variables. This plot as well as frequency counts and risks are updated automatically whenever any action that modifies key variables is applied. We note that the interpretation of the mosaic plot is mostly helpful for expert users.



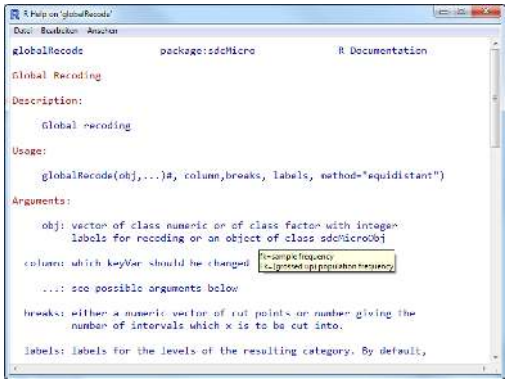
(a) Original distribution of *age* (continuous).



(b) Variable *age* recoded into age groups and converted to a factor.



(c) Frequency counts and individual risks of all combinations of categorical key variables.



(d) help file for global recoding.

Figure 5: The global recoding interface. All key variables can be recoded.

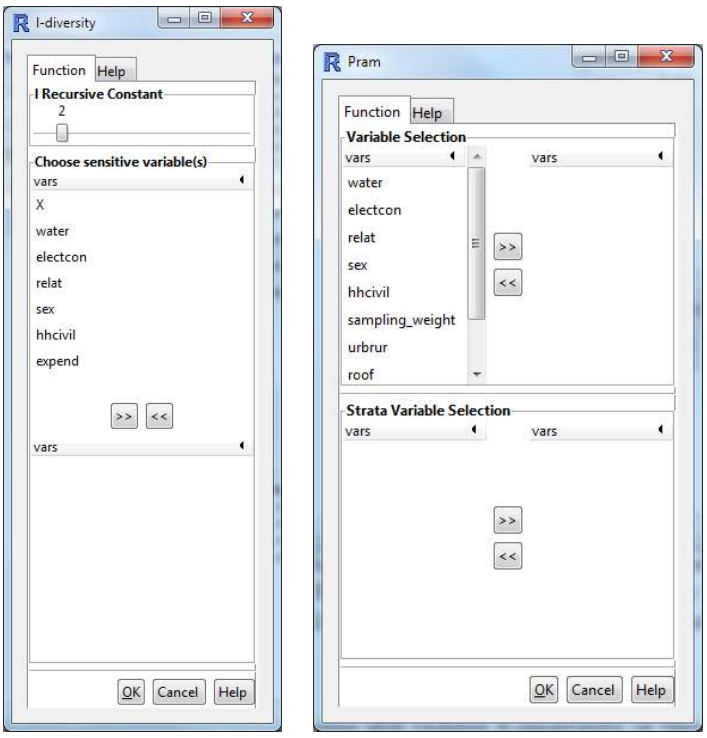
6.2 PRAM

After selecting the post randomization method, PRAM, a new window pops up, in which the user can select variables to apply the PRAM method to (see Figure 6(b)). PRAM swaps categories randomly with predefined probabilities. Optionally, a variable for stratification can also be selected. If a stratification variable has been selected, PRAM is applied on each stratum independently. Once the procedure has finished, a window summarizing the results pops up. The output of this window consists containing a table with three columns. The first column displays the variable names; the second column ("nrChanges") displays the total number of changed values; and the third column ("percChanges") displays the percentage of changed values for each variable.

6.3 Local Suppression

Even after recoding key variables, some combinations of characteristics of these variables may still violate k-anonymity, or some observations may still have relatively high individual disclosure risks. Further recoding, however, may not be possible because the data utility would be too low. At this stage, local suppression can be applied. Two methods are available in `sdcMicroGUI`.

The first method applies optimal local suppression with the goal of reaching k-anonymity when the button Local suppression (optimal - k-anonymity) is clicked.



(a) l-Diversity.

(b) Post randomization method (PRAM).

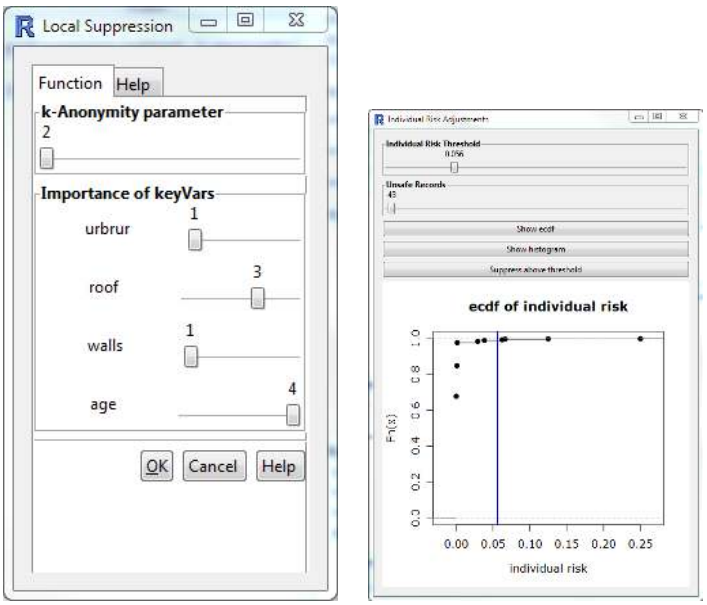
Figure 6: l-diversity and PRAM.

In this window, the user can choose the importance of variables for the local suppression algorithm. This means that the higher the rank or importance of a variable, the higher the probability that required suppression is applied to this variable. If one has specified a key variable that is considered extremly important (e.g region) one may specify the lowest importance for this variable. Thus, values in this variable will only be suppressed if no other possibility is avaiable to reach k-anonymity. Otherwise, the choice of importance of key-variables can depend for example on the future use of the anonymized data set.

The probabilities of applying suppressions are lower for variables with lower ranks. `sdcMicroGUI` automatically suggests an optimal order of importance (based on the number of unique values for each key variable), as shown in Figure 7(a).

By adjusting a slider, the user may also change the parameter k (how often each key should exist in the data set) for k-anonymity, which is typically 3 or 4. After the procedure has finished, the resulting number of combinations of the key variables violating k-anonymity (which is usually zero) is automatically updated and printed at the top left of this tab, together with the updated number of new suppressions.

Another option is to apply local suppression only to specific variables by clicking on the button *Local Suppression (threshold - indiv.risk)*; see Figure 7(b). Changing the slider "Individual Risk Threshold" sets a risk threshold. This threshold is simply an upper limit for individual risks that is relevant for the value of "Unsafe Records". This value is automatically updated and shows the number of observations that currently have larger individual risks than the selected risk threshold. By clicking on buttons *Show ecdf* or *Show histogram*, the user is presented with an interactive plot window, which displays the empirical cumulative density function of the individual risks or a histogram. In both cases, a blue vertical line is displayed



(a) Optimal local suppression based on k-anonymity.

(b) Local suppression based on risk threshold.

Figure 7: Optimal and individual local suppression.

at the current value of the individual risk threshold. If the button *Suppress above threshold* is clicked, the use can select one of the key variables. Finally, all values in this variable for observations having higher individual risks than this threshold will be suppressed. This is a simple method to reduce individual risks because it is applied only to those observations that already have risks over a given threshold value. For example, if the risk threshold is chosen as 0.1 and the variable is chosen as *age*, the age for all observations that have an individual re-identification risk larger than 0.1 will the suppressed.

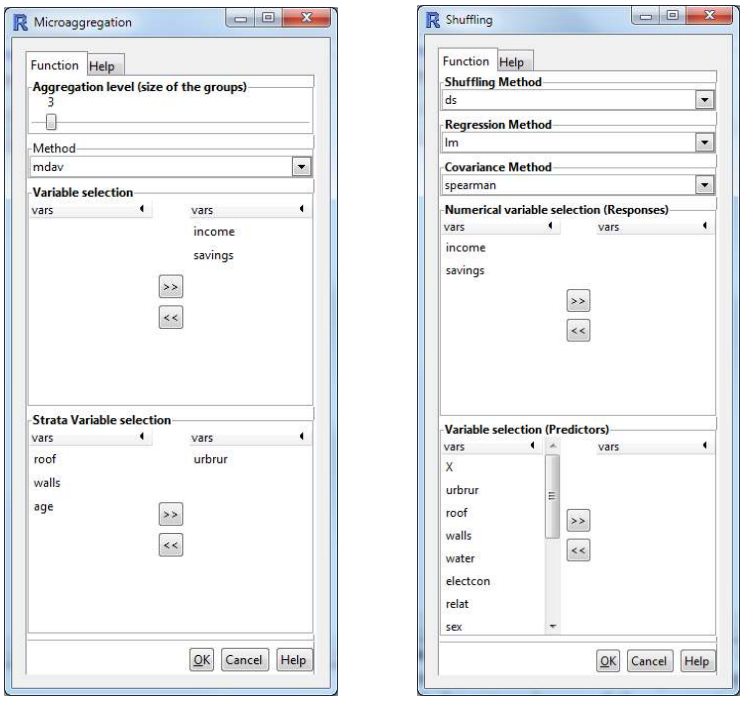
7 Anonymisation of Continuous Key Variables

The Continuous tab of the main window applies SDC methods for continuous variables and displays risk measures and measures of information loss, as discussed in Section 3.2. After applying any disclosure limitation technique, disclosure risks and data utility measures are automatically re-calculated and updated values are printed in this tab, providing information on how the continuous scaled key variables are preserved and how large the disclosure risk remains. The methods Microaggregation, Adding Noise and Shuffling can be selected in this window.

7.1 Microaggregation

By clicking on the button *Microaggregation*, a new window pops up, as shown in Figure 8(a). The user selects an aggregation level by moving a slider, a microaggregation method in a drop-down box and at least one numeric key variable. Five micro-aggregation methods are available in sdcMicroGUI: *"mdav"* [see, e.g., Domingo-Ferrer and Mateo-Sanz, 2002], *"rmd"* [Templ and Meindl, 2008], *"pca"* [see, e.g., Templ, 2008], *"clustpppca"* [Templ, 2008] and *"influence"* [see, e.g, Domingo-Ferrer et al., 2002]. It is also possible to apply micro-aggregation to subsets of the data separately. If this option is used, the user has to select an additional strata-

variable that defines the partition of the dataset. A help tab is also available in this window where more information about possible methods and parameters is available.



(a) Window to specify parameters and options for micro-aggregation of continuous variables.

(b) Specifying options for shuffling method.

Figure 8: The Micro-aggregation and Shuffling windows.

7.2 Adding Noise

The user can add stochastic noise to numerical key variables by clicking on *Add Noise*. In this case, a new window pops up, where the user specifies whether to add additive or correlated noise [Brand, 2004] using a drop-down menu. The user must also specify the desired amount of noise, in percentages, and select at least one numeric key variable. If the user clicks on *OK*, the selected method is applied to the chosen variable(s). It is, however, always possible (as it is in all windows of the GUI) to cancel the current operation by clicking on *Cancel*. As in the pop-up window for micro-aggregation, a help tab is available in this window as well, providing additional information.

7.3 Shuffling

To anonymize continuous key variables, shuffling [Muralidhar and Sarathy, 2006] can be selected by clicking on button *Shuffling*. A new window will open automatically, and the user can select the shuffling method, regression method, and covariance method, respectively, using drop-down menus.

The default shuffling method is *ds* [Muralidhar and Sarathy, 2006], but *mvn* [Templ et al., 2013b,a] and *mlm* [Templ et al., 2013b,a] may also be selected. The default regression method is *lm* (linear regression). As a robust alternative,

method MM [Maronna et al., 2006, Templ et al., 2013a] may be chosen; in this case, a robust regression with M-estimator is used. Another choice for the covariance method includes *spearman*. Alternatives that can be selected are *pearson* and the robust variant *mcd* [Rousseeuw and Van Driessen, 1999].

Afterwards, users must select response and predictor variables. In `sdcMicroGUI`, all variables selected as predictors are used without any interactions between them. Any complex formula can be applied using function shuffle () from `sdcMicro` [Templ et al., 2013b]. Note that all variables can serve as predictors, which means that this selection is not limited to previously selected key variables. As in the other pop-up windows, additional help is provided in the help tab (see Figure 8(b)).

8 Exporting Results

8.1 Export Anonymised Datasets

8.1 Export Anonymized Datasets Using *Data* → *Export* at the top of the main menu, it is possible to export the anonymized dataset into various formats. By clicking on the appropriate menu entry, the data can be exported as plain text `.csv` files, as well as in formats that can be read by other statistical software such as `SAS`, `SPSS` or `Stata`. In addition, the dataset can be saved directly to the R workspace or by using the R binary format.

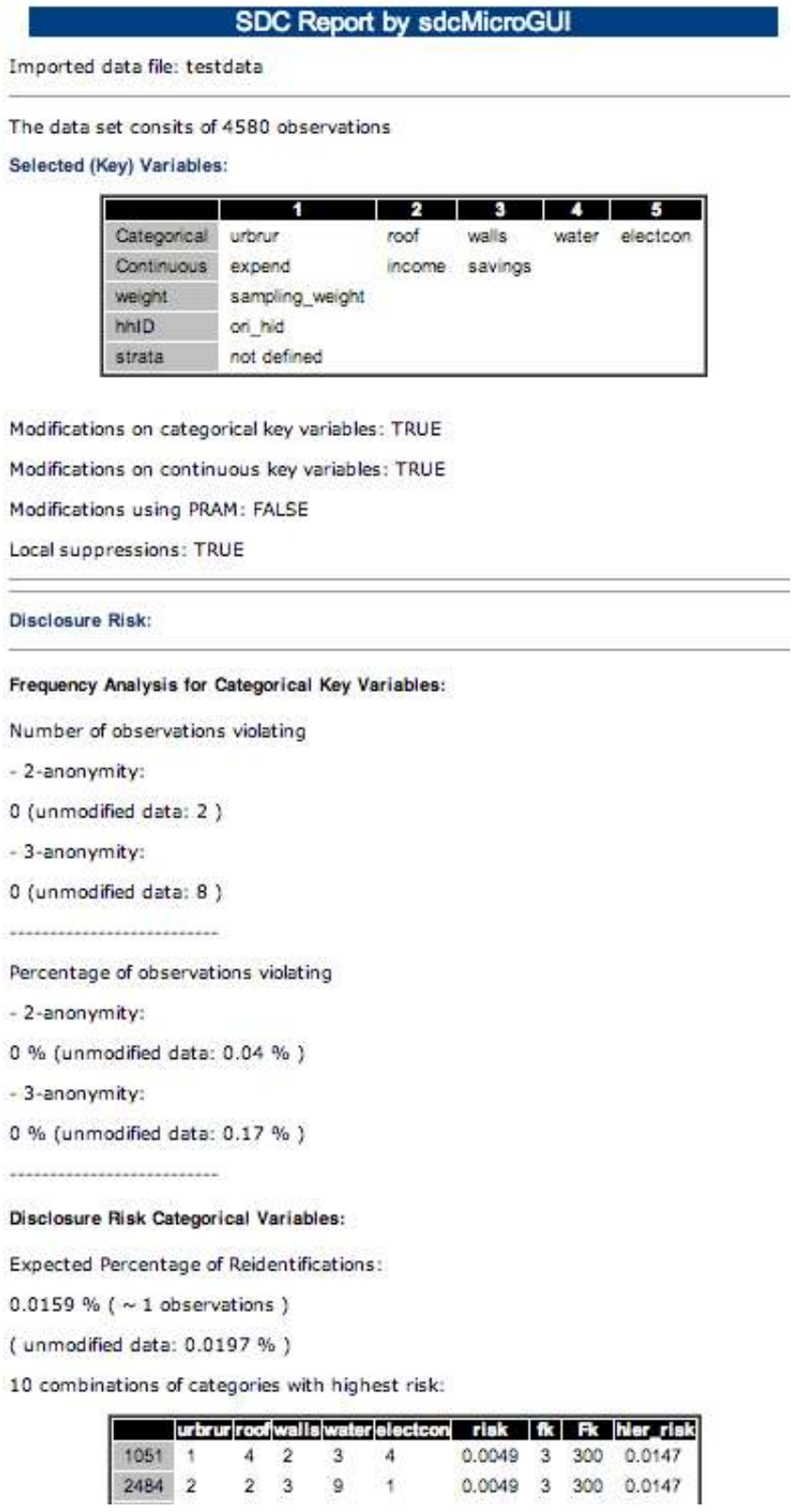
8.2 Reports

Selecting *Data* → *Generate Report* in the top menu opens a new window from which two different reports, internal and external, can be produced by selecting the corresponding button. It is also possible to select the output format. Reports can be saved as html, pdf or plain text files. A sample output is shown in Figure 9.

Internal Report by setting function argument *internal=TRUE* include information about the performed actions, disclosure risk, measures of information loss and session information on the software versions used. This detailed report is suitable for the organization that holds the data for internal use and documentation of the anonymization procedure.

External Report by setting function argument *internal=FALSE* include less information than internal reports. For example, all information on disclosure risks and information loss is suppressed. This report is suitable for external users of the anonymized data.

Figure 9 shows the first page of the internal report, where information on selected variables, anonymization methods applied and disclosure risk is displayed. Detailed analysis on risk and utility follows. Information included in the report always depends on the anonymization process. For example, if PRAM is not applied, no specific summary for variables subjected to PRAM is available. But if PRAM is used, the entire disclosure risk summary is presented differently. We note that the report gives a summarization of the anonymization process and helps a lot to document the differences between different anonymization approaches. However, it still only summarizes results that are conveniently available in the graphical user interface at all times.



SDC Report by sdcMicroGUI

Imported data file: testdata

The data set consits of 4580 observations

Selected (Key) Variables:

	1	2	3	4	5
Categorical	urbrur	roof	walls	water	electcon
Continuous	expend	income	savings		
weight	sampling_weight				
hhID	ori_hid				
strata	not defined				

Modifications on categorical key variables: TRUE

Modifications on continuous key variables: TRUE

Modifications using PRAM: FALSE

Local suppressions: TRUE

Disclosure Risk:

Frequency Analysis for Categorical Key Variables:

Number of observations violating

- 2-anonymity:

0 (unmodified data: 2)

- 3-anonymity:

0 (unmodified data: 8)

Percentage of observations violating

- 2-anonymity:

0 % (unmodified data: 0.04 %)

- 3-anonymity:

0 % (unmodified data: 0.17 %)

Disclosure Risk Categorical Variables:

Expected Percentage of Reidentifications:

0.0159 % (~ 1 observations)

(unmodified data: 0.0197 %)

10 combinations of categories with highest risk:

	urbrur	roof	walls	water	electcon	risk	fk	Fk	Nier_risk
1051	1	4	2	3	4	0.0049	3	300	0.0147
2484	2	2	3	9	1	0.0049	3	300	0.0147

Figure 9: A screenshot of the first page of the automatic generated SDC-report.

9 The Script: Reproducibility of Results Obtained with the GUI

Any result obtained from clicking and setting parameters interactively in `sdcMicroGUI` is reproducible. This is a major feature of the software because every action the user performs is internally stored, saved and listed and can be reviewed in the *script frame*. To access this window, select Script *Script* → *View* on the main menu; see Figure 10.

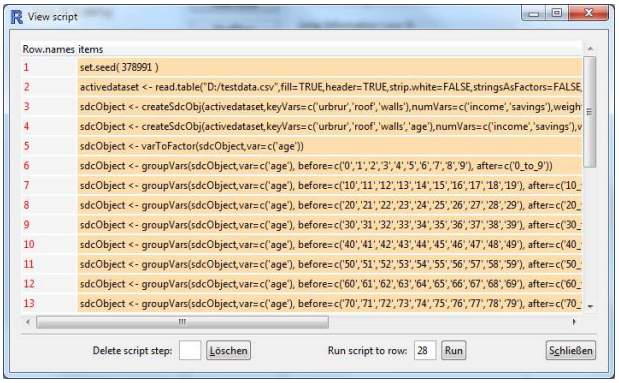


Figure 10: The view script window showing the anonymization history

.

It is possible to not only view the current script, but also to export (*Script* → *Export*) and import (*Script* → *Import*) scripts from `sdcMicroGUI`. Therefore, it is easy to reproduce previously produced output. It is even suitable to modify some steps or alter the output. Users can also remove specific steps from the script when navigating through it or execute steps only up to a certain point. This feature is very helpful for reproducing older results, continuing work that was previously started or restarting steps of the anonymization quickly.

10 Working with the sdcMicro Package

For each method, we show its usage in the software via command lines, using `sdcMicro`. We start with a brief introduction to the package before the methods are explained.

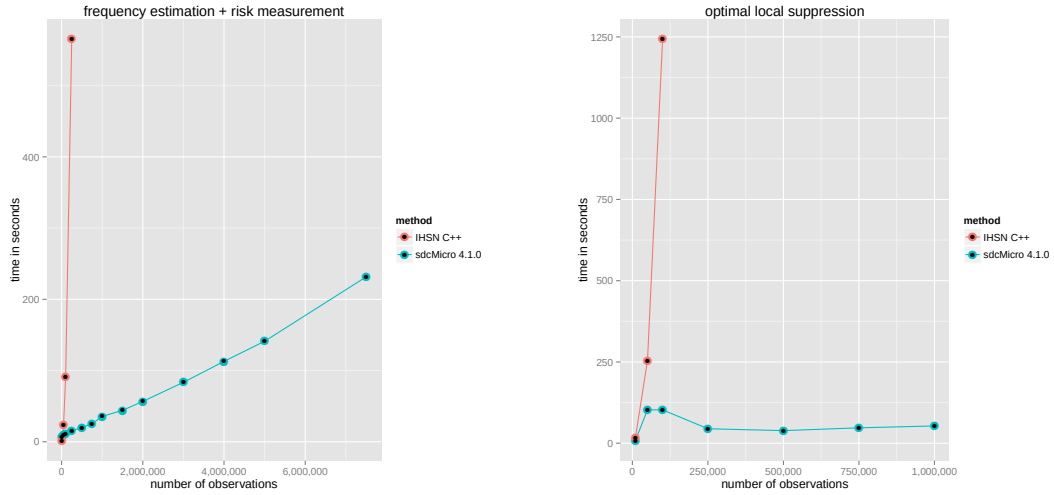
10.1 General Information about `sdcMicro`

In the last few years, the statistical software environment R (R Development Core Team, 2011, also known as R) has become more popular. R currently has more users than any other statistical software[3], and is the standard statistical software for data analysis and graphics. For statisticians, R has become the major programming language in its field.

Version 1.0.0, the first version of the `sdcMicro` package, was released in 2007 on the comprehensive R archive network (CRAN, http://cran.r-project.org). The

[3]See, for example, http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html, where `R` entered the top 20 of all programming software in January 2012. `SAS` is ranked on place 32.

current release, version 4.6.1, is a huge step forward. Almost all methods are implemented in a highly object-oriented manner (using S4 classes) and have been written internally in `C` or call `C++` code, which allows for high-performance computations. The International Household Survey Network (IHSN) provided `C++` code for many methods that were partly integrated into sdcMicro and partly rewritten. One example is given in Figure 11, where we show the computation time of the current version of sdcMicro (version 4.1.0) compared to the previous implementation in sdcMicro (< version 4.1.0) that calls the IHSN `C++` code. While the IHSN `C++` solutions were exponential in computation time regarding the number of observations, the new implementation has linear complexity (see Figure 11(a)). For special tasks (e.g., heuristic optimal local suppression), the computation time may even provide faster growth, i.e., less than linear growth. The higher the number of observations, the higher the probability that k-anonymity is reached. This fact is used internally for optimizing the calculations.



(a) Frequency estimation and risk measurement.

(b) Local suppression

Figure 11: Computation time of IHSN `C++` code (sdcMicro version $<$ 4.1.0) and sdcMicro (version $\geq$ 4.1.0).

After installing and starting `R`, the package description, which shows summary information about the package, can be called by typing the following:

```
packageDescription("sdcMicro")
```

10.2 Getting Help

For each of the methods implemented in `sdcMicro`, a help file is available, which not only describes all possible parameters that can be changed, but also features simple, working examples that can be directly copied into `R`. The help file for a given function can be accessed by calling an `R`-function with a `?` directly before the function name.

For example, the following commands call the index of methods available in the package sdcMicro, and the help file for the micro-aggregation function:

```
help(package=sdcMicro)   # index of methods
?microaggregation        # same as help("microaggregation")
```

sdcMicro features vignettes, which are manuals available in pdf formats. These vignettes contain interesting information, including the most recent version of this tutorial. The following command browses the available vignettes of sdcMicro.

```
vignette(package="sdcMicro")
```

10.3 S4 Class Structure

The sdcMicro package supports both the straightforward application of methods to data and the application of methods to *sdcMicroObj*. For example, when applying micro-aggregation on three continuous key variables to the dataset testdata, the command **microaggregation**(testdata[,**c**("expend","income","savings")]) is equivalent to **microaggregation**(sdc), if sdc has been properly defined as an object of class *sdcMicroObj*.

To start, first load the sdcMicro package installed (installation instructions have already been given in Section 2, by typing the following:

```
require(sdcMiro)
```

To define an object of class *sdcMicroObj*, the function **createSdcObj**() can be used. In this case, all required parameters have to be specified, including, for example, categorical and continuous key variables, the vector of sampling weights and optionally stratification and cluster IDs. The following example shows how to generate such an object using the test data included in library sdcMicro.

```
load(testdata)
sdc <- createSdcObj(testdata,
 keyVars=c('urbrur','roof','walls','water','electcon','sex'
    ),
 numVars=c('expend','income','savings'),
 w='sampling_weight', hhId='ori_hid')
```

The following shows how to define the categorical and continuous key variables, vector of weights and household IDs; the slots of the *sdcMicroObj* sdc are pre-filled:

```
> slotNames(sdc)

 [1] "origData"          "keyVars"           "pramVars"
 [4] "numVars"           "ghostVars"         "weightVar"
 [7] "hhId"              "strataVar"         "sensibleVar"
[10] "manipKeyVars"      "manipPramVars"     "manipNumVars"
[13] "manipGhostVars"    "manipStrataVar"    "originalRisk"
[16] "risk"              "utility"           "pram"
[19] "localSuppression"  "options"           "additionalResults"
[22] "set"               "prev"              "deletedVars"
```

The first slot contains the original data, the second slot contains the index of categorical key variables, and so on. For details, type **help**("createSdcObj") into R).

Every method is then applied on the *sdcMicroObj* and all related computations are done automatically. For example, the individual risks are re-estimated whenever a protection method is applied, and then the corresponding slots are updated. In addition, the system knows which methods can be applied to which variables.

When applying a method that is suitable for categorical variables, the user does not have to specify the variables again.

The application of a method to an object of class *sdcMicroObj* is done by `method(sdcMicroObj)`, where the `method` is a placeholder for any method available in `sdcMicro`. Below is an example of this object-oriented implementation approach. In this example, the method micro-aggregation is applied on an object of class sdcMicroObj. Since micro-aggregation is suitable only for continuously scaled key variables, the categorical variables remain untouched. Additionally, the risk and utility slots are updated to contain the new estimates using current values of the micro-aggregated variables. In this example, default values for parameters are used, but it is possible to change the default values. For details, see **help**("microaggregation"). The following code shows the application of a microaggregation to an object of class *sdcMicroObj*:

```
sdc <- microaggregation(sdc)
```

The slots of the *sdcMicroObj* can be accessed also using function **get**.sdcMicroObj (), as shown below.

```
get.sdcMicroObj(sdc, "utility") ## access utility
get.sdcMicroObj(sdc, "keyVars") ## access cat. key
    variables
```

Print methods are available to show the relevant information. See the following code listing for printing the risk:

```
print(sdc, "risk")
```

```
Risk measures:

Number of observations with higher risk than the main part of the data: 0
Expected number of re-identifications: 4.33 (4.65 %)

Information on hierarchical risk:
Expected number of re-identifications: 5.92 (6.37 %)
----------------------------------------------------------------------
```

More information on `sdcMicro` and its facilities can be found in the manual of `sdcMicro`; see Templ et al. [2013b].

References

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

J. Domingo-Ferrer and J.M. Mateo-Sanz. Practical data-oriented microaggregation for statistical disclosure control. *IEEE Trans. on Knowledge and Data Engineering*, 14(1):189–201, 2002.

J. Domingo-Ferrer, J.M. Mateo-Sanz, A. Oganian, and A. Torres. On the security of microaggregation with individual ranking: analytical attacks. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):477–492, 2002.

L. Franconi and S. Polettini. Individual risk estimation in μ-Argus: a review. In J. In: Domingo-Ferrer, editor, *Privacy in Statistical Databases, Lecture Notes in Computer Science*, pages 262–272. Springer, 2004.

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. 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.

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

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

P.J. Rousseeuw and K. Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41:212–223, 1999.

P. Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

M. Templ. Statistical disclosure control for microdata using the R-package sdcMicro. *Transactions on Data Privacy*, 1(2):67–85, 2008. URL http://www.tdp.cat/issues/abs.a004a08.php.

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, A. Kowarik, and B. Meindl. Introduction to statistical disclosure control (sdc). Project: Relative to the testing of SDC algorithms and provision of practical SDC, data analysis OG, 2013a.

M. Templ, A. Kowarik, and B. Meindl. *sdcMicro: Statistical Disclosure Control methods for the generation of public- and scientific-use files. Manual and Package.*, 2013b. URL http://CRAN.R-project.org/package=sdcMicro. R package version 4.1.1.