IHSN GUI Tutorial for sdcMicroGUI (and sdcMicro)

Matthias Templ, Bernhard Meindl and Alexander Kowarik

Vienna, November 5, 2013

http://www.data-analysis.at

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

1 An Overview of sdcMicroGUI

The sdcMicroGUI [Kowarik et al., 2013] serves as an easy-to-handle 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. It is possible for the user in sdcMicroGUI to interact with microdata in an interactive way. The software performs automated recalculation and display of frequency counts, individual and global risk-measures, information loss and data utility after any anonymization step. Changes to risk and utility measures of the original data are also conveniently displayed in the GUI (graphical user interface). 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. The software helps to reproduce any results.

In this guidelines, the concept of sdcMicroGUI and its possibilities are presented. The sdcMicroGUI package is an highly interactive tool which takes into account the following aspects:

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

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

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

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

Interactivity: Risk und utility measures are automatically estimated whenever user apply a disclosure limitation technique. The corresponding numbers are instantly shown in the corresponding parts of the GUI. Users immediately sees the effect of any action. In addition, the risk and utility of the original unmodified data are displayed, which guides somehow the user on the effectiviness of the anonymisations

Undo Button: Users can undo the last step done in the GUI. This is a very useful feature since the undo-feature makes it possible to try out methods with different parameters, get instant feedback and go back and forth until the result is satisfying. Currently it is possible to go back exactly one step in the history.

Report: Directly from the user interface it is possible to produce automatically generated, standardized reports in various output formats. The report can be exported to html, latex or plain text-files. Users can select to generate two different type of reports. The more detailed (internal) version includes the anonymisation methods applied but also estimates of risk and utility are reported depending on the methods used (different output for different methods). The shorter (external) version returns a brief summary on the anonymisation procedure that have been applied. This output is also suitable for external viewers. In this report-type, detailed comparisons and summaries are suppressed and are not included in the report. For more information, see Section 6.2.

Reproducibility: `sdcMicroGUI` offers the possibility to 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 anonymisation procedure can be reproduced either simple by loading a script into the GUI or by pasting the script directly into an `R`-console.

1.1 Key Elements of the Graphical User Interface

The main menu

In Figure 1 parts of `sdcMicroGUI` are shown. Specifically, in Figure 1(a) the top-menu is shown while in Figure 1(b) the import mask for *.csv*-files is shown. We now continue to describe the elements of the main menu that is located at the top of the GUI and that is always visible.

- **GUI**: this menu-item allows to close or restart the interface as well as to check for possible updates of `sdcMicro` and/or `sdcMicroGUI`.
- **Data**: in this menu it is possible to import data (from various data formats) as well as existing `R`-objects into the GUI. It is also possible to export data (also to different data formats) and generate a report from here.
- **Script**: the script menu item allows to save, load or view a script that was generated by `sdcMicroGUI`.
- **Help**: the help-section gives access to different resources such as information on possible disclosure control methods or the documentation of the underlying functions from package `sdcMicro`.
- **Undo**: The button ⟶ *Undo* menu entry allows to go back one step in the anonymisation process. It is possible to try out an anonymisation method and if the results are not satisfying this last action can be reverted easily.

The main window of the GUI

The GUI generally displays three tabs, see also Figures 5(a), 5(b) and 5(c).

Identifiers Shows a summary of the current selection of key

In this view, direct identifiers can be removed by clicking on the corresponding button. In this tab it is also possible to reset the current choice of key-variables, see also Figure 5(a)).

Categorical: This tab is divided into three parts. On the left hand In the middle part, the methods that can be used for the the anonymisation of categorical key variables (Recoding, PRAM, local suppression) are displayed and can be selected and applied as shown in Figure 5(b). On the right hand side some important measures on information loss are shown. At the top, information about recoding (number of categories, mean size and number of observations in the smallest category) for all key-variables (for both original and modified) are listed. At the bottom of the right hand side, the number of suppressions within each key-variable is printed.

Continuous: The third tab is also divided into three parts. On the In the middle part, the methods (microaggregation, adding noise, shuffling) that can be used to anonymize continuous key variables are shown and can be selected and applied. On the right hand side two measures of information loss (IL1 and differences in eigenvalues) are printed.

All frames and views in the GUI for presenting 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 like recoding, PRAM or local suppression get 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 and show the current values automatically.

1.2 Installation and Updates

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

Install R: If you already have R installed, make sure that you are using the current version. If the software is not installed on your computer, 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 when installing the software.

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

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

The installation is only necessary once. We note that the graphical user interface depends on GTK+ to draw windows. When installing sdcMicroGUI all dependencies (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. Using sdcMicroGUI it is also possible to check for an updated version by clicking on the menu-item *GUI → Check for Updates* which should be done regularly.

If your organisation use a proxy server to connect to the internet, automatic access of R is usually restricted. However, a simple trick gives you the necessary internet connection from within R. In case you have a proxy server just type

```
setInternet2(TRUE)
```

into the R-Console. Afterwards you are able to install packages.

(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.

2 Open the GUI

Open the software R and type:

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

This will load the `sdcMicroGUI` package into R and calls the point an click graphical user interface. If you have not installed `sdcMicroGUI`, you will see an error message and should follow the steps to install the package as described in Section 1.2.

2.1 Step 2: Select, Load or Import Data

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

Very important is the advanced functionality which is available to import text-delimited *.csv* files, see Figure 1(b). In this case, the user is presented with a data preview window (see again Figure 1(b)) that shows the first rows of a data set with the current data import parameters and several clickable checkboxes to change import options such as:

- *header*: does the the first line of a data set contain column names
- *fill*: if checked, blanks are added for rows with unequal length
- *strip white*: allows the stripping of leading and trailing white space 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 seperator between values, the decimal operator, quotes, skip and the coding of missing values (NA-strings) can be specified. As soon the user changes a field, the preview-window of the data changes according to the options and informs the user if the data get correctly imported. Note that also the type of each variable can be specified (*numeric* or *factor*[1]) when importing a *.csv*-File using the button ⟶ *Adjust Types* as shown at the bottom in Figure 1(b).

3 Selecting Key Variables

After a data set was selected or imported, a new window - the variable selection frame - pops up and the buttons ⟶ *Select key variables / Reset* and ⟶ *Remove direct identifiers* in the first tab of the GUI interface become active. An important part in the anonymisation of data is the variable selection, see Figure 3. Statistical methods and the corresponding functions in R are in most cases specific in terms of the scale of variables. Some functions should only be applied to categorical (in R these variables correspond to vectors of class *factor*) variables while some are suitable for continous (in R they are vectors of type *numeric*) variables only.

In any case, only **categorical** and **continuous key variables** should be selected in this frame because any other categorical or continuous variables of the data set do not be contribute in any aspect to the anonymisation process. Only for shuffling, all variables are alaways made selectable (as predictors), independently of this choice.

Figure 2: Defining strata window.

Some functions can also be applied on **domain level**. In this case, it is required to specify or create a variable defining population subgroups. To create a new strata variable (for example a combination of several categorical variables), the strata selection window can be used. This window can be accessed by clicking on the button ⟶ *Generate Strata Variable* in the variable selection window, see Figure 2.

Also information on **clustering** (for example households) are often required to be specified and can be taken into account by using the variable selection window. Furthermore it might also be important to select a **weight vector**, especially if the microdata have been collected from a complex survey. This kind of information has to be provided by the user, otherwise the system can not make use of this knowledge. Note, that most of the variable selection options are optional. For

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

example, users are not forced to select any continuous (in the GUI: *numerical*) variables if they are not present in the data.

As mentioned before, it is often necessary to generate a new stratification variable that is a combination of 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 2. In the case that variables have the wrong scale (for example if categorical key variables are saved as type *numeric*), the global recoding frame automatically pops up. In the global recoding window it is then possible recode the corresponding variables.

Figure 3: The variable selection frame of the GUI.

4 Anonymisation of Categorical Key Variables

4.1 Recoding

Clicking on the button *Recode* in the tab *Categorical* of the main window opens a window in which the categorical key variables can be recoded, as it is shown in Figure 4.

It is possible to recode all categorical key variables separately. The corresponding variable names are visible in the menu of the key variables configuration frame. Any variable can now be converted, recoded, grouped and renamed. For converting continuous scaled variables into classes, breaks and label names can be specified. It is also possible to group categories of factors into broader categories (using the button ⟶ *Group selected level*) and also rename specific categories by clicking on button ⟶ *Rename selected level* after having selected a factor level. The distribution of the variables is always shown graphically as well as information on the tabulated variable. Additionally, the frequency counts of all key variables are available and visible in a separate tab.

Figure 5(a) shows the the graphical user interface after the necessary variables have been selected and optionally recoded. As it has already been mentioned, the user interface consists of three main parts which are organised in tabs and have

(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 4: The global recoding interface. All key variables can be recoded.

already been described in Section 1.1.

4.2 Frequency Counts and Disclosure Risk

Sample- and population frequencies (f_k and F_k, respectively) are visualized when tab ⟶ *Frequencies*) in the global recoding menu is clicked. But not only the frequencies but also the individual risks [Franconi and Polettini, 2004] and the values of the categorical key variables are printed. The table containing these statistics is interactive and sortable. This means that for example clicking on the top of column *risk* 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.

In tab ⟶ *Mosaic Plot* of the global recoding menu the mosaic plots of all selected key-variables is shown. The plot as well as the frequency counts and risks are updated automatically whenever any action that modifies key-variables is applied. Similar information is available in the second tab (*Categorical*) of the `sdcMicroGUI` main window, see Figure 5(b). Additionally, all observations violating 3-anonymity are directly accessible. These observations are shown if the user clicks on the the button ⟶ *View Observations violating 3-anonymity*. In order to compare the impact of anonymisation methods already applied, the same

(a) View of Tab ”Identifiers”

(b) View on Tab ”Categorical”

(c) View on Tab ”Continuous”

Figure 5: Showing the three main tabs of the GUI. Information on the content was already given in 1.1.

information calculated with the original, unmodified data is also displayed.

Observations having high individual risks can be viewed by clicking the button ⟶ *View Observations with high risk*. Information on how many observations are expected to be re-identified under the given selection of key-variables is displayed as well as information on the number of observations with considerable higher risk than the main part of all observations is shown.

Last but not least, the l-diversity measure [Machanavajjhala et al., 2007] can be calculated. By clicking on the corresponding button a new window pops up in which the user can select sensitive variables and set the l-recursive constant (see Figure 6(a)). As soon as the button ⟶ *OK* in this window is clicked, another window containing the results pops up.

On the right hand side (see Figure 5(b)) information about the effect of recoding in key-variables are printed. Also, the number and percentages of suppressions for each categorical key variable is shown. The information about the frequencies, k-anonymity [Sweeney, 2002] and individual risks is always updated whenever a method is applied to categorical key variables. Thus, users get an impression how the recodings infect the frequency counts and individual risks.

As shown in Figure 5(b) methods for anonymisation of categorical key variables can be selected clicking on corresponding buttons in the main window of the graphical user interface. This includes the already discussed global recoding facilities, PRAM [Gouweleeuw et al., 1998] and two different methods to perform local suppression.

(a) l-Diversity.

(b) Post randomization method (PRAM).

Figure 6: l-diversity and PRAM.

If method PRAM - the post randomisation method that swaps categories randomly with predefined probabilites - is selected, a new window pops up. In this window the user has to select variables that should be pramed (see Figure 6(b)). Moreover, a variable for stratification can also be selected. In this case, PRAM is applied on each strata independently.

4.3 Local Suppression

Often it is the case that even after recoding key-variables, some combinations of characteristics of these variables still violate k-anonymity or that some observations still have relatively high individual disclosure risks. However, it might be the case that further recoding is not possible because either the data utility would be too low. At this stage, local suppression can be applied. Two methods are available in `sdcMicroGUI`. The first one applies local suppression in an optimal manner with the aim to reach k-anonymity.

Figure 7(a) shows the window that is opened if the user wants to perform optimal local suppression after clicking on button ⟶ *Local suppression (optimal - k-anonymity)*. In this window the user can choose the importance of variables for the local suppression algorithm. This means, that the higher the rank (importance) of a variable is, the higher is the probability that required suppression are applied to this variable. The probabilities to get suppressions are lower for variables with lower ranks. `sdcMicroGUI` automatically suggests an optimal order of the importance as it is shown in Figure 7(a).

By adjusting a slider, the user may also change the parameter k 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 zero most of

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

(b) Local suppression based on risk threshold.

Figure 7: Optimal and individual local suppression.

the time) is automatically updated and printed in the top left of this tab together with the updated number of (new) suppressions.

Another option is to apply local suppression can be also to specific variable only by clicking on the button ⟶ *Local Suppression (threshold - indiv.risk)*. In this case, risky observations can be specified interactively using a slider as it is shown see Figure 7(b). After setting a threshold of individual risks, all values of a variable (that has to be specified by the user) are suppressed if the current individual risk of this observation is higher than the selected threshold value.

5 Anonymisation of Continuous Key Variables

On the second tab ⟶ *Continous* of the main window, sdc-methods for continous variables can be applied and risk-measures and measures of information loss are displayed as it was already discussed in section 1.1. After applying any disclosure limitation techique, the disclosure risks and the data utility measures are automatically re-calculated and updated values are printed in this tab. Thus, users get an impression on both, how the continuous scaled key variables are preserved and how large the disclosure risk remains. The following methods can be selected in this window:

5.1 Microaggregation

By clicking on button ⟶ *Microaggregation*, a new window pops up as it is shown in Figure 8(a). In this window the user has to select an aggregation level by moving a slider, select a microaggregation method using a drop-down box and select at least one numeric key variable. Optionally, it is also possible to apply microaggregation to subsets of the data separately. If this option is wanted, the user has to select an additional strata-variable that define the partition of the data set. In this window also a help tab is available in which more information about possible methods and parameters is available.

(a) Window to specify parameters and options for microaggregation of continuous variables.

(b) Specifying options for shuffling method.

Figure 8: The microaggregation and the shuffling window.

5.2 Adding (Correlated) Noise

The user can add stochastic noise to numerical key-variables by clicking on button ⟶ *Add Noise*. In This case, a new window pops up in which the user has to specify if he wants to add additive or correlated noise [Brand, 2004] by selecting the appropriate method using a drop-down menu. The user also has to specify the desired amount of noise (in percentage) and select at least one numeric key-variable. If the user clicks on the button ⟶ *OK*, the selected method is applied to the chosen variable(s). It is however always possible (as it is in all windows of the graphical user interface) to cancel the current operation by clicking on button ⟶ *Cancel*. As in the pop-up window for microaggregation, also in this window a help-tab is available providing additional information.

5.3 Shuffling

Shuffling [Muralidhar and Sarathy, 2006] can be selected to anonymize continuous key-variables by clicking on button ⟶ *Shuffling*. Various methods are available but method *ds* [Muralidhar and Sarathy, 2006] is selected by default. After clicking on the button, a new window opens automatically. In this window the user can select the method for shuffling, the regression method and the covariance method using drop-down menus. Afterwards, variables have to be selected as response- and predictor variables. In the implementation of `sdcMicroGUI`, all variables selected for acting as predictors are used without any interactions between them. Any complex formula can be applied using the shuffling function from `sdcMicro`. Please also 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).

6 Exporting Results

6.1 Export anonymised data-sets

Using *Data* → *Export* in the main-menu on the top, it is possible to export the anonymized data-set 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 data set can be saved directly to the R workspace or using the R binary format.

6.2 Reports

Selecting *Data* → *Generate Report* in the top menu opens a new pop window from which two different kind of reports can be produced by selecting the corresponding radio button. Also it is possible to select the output format. The reports can be saved as html-, pdf- or plain-text files. A sample output is shown in Figure 9.

Internal Report: includes information about the performed actions, the disclosure risk and measures of information loss and a session info on the software versions used. This detailled report is suitable for the organisation that holds the data for internal information and documentation of the anonymisation procedure.

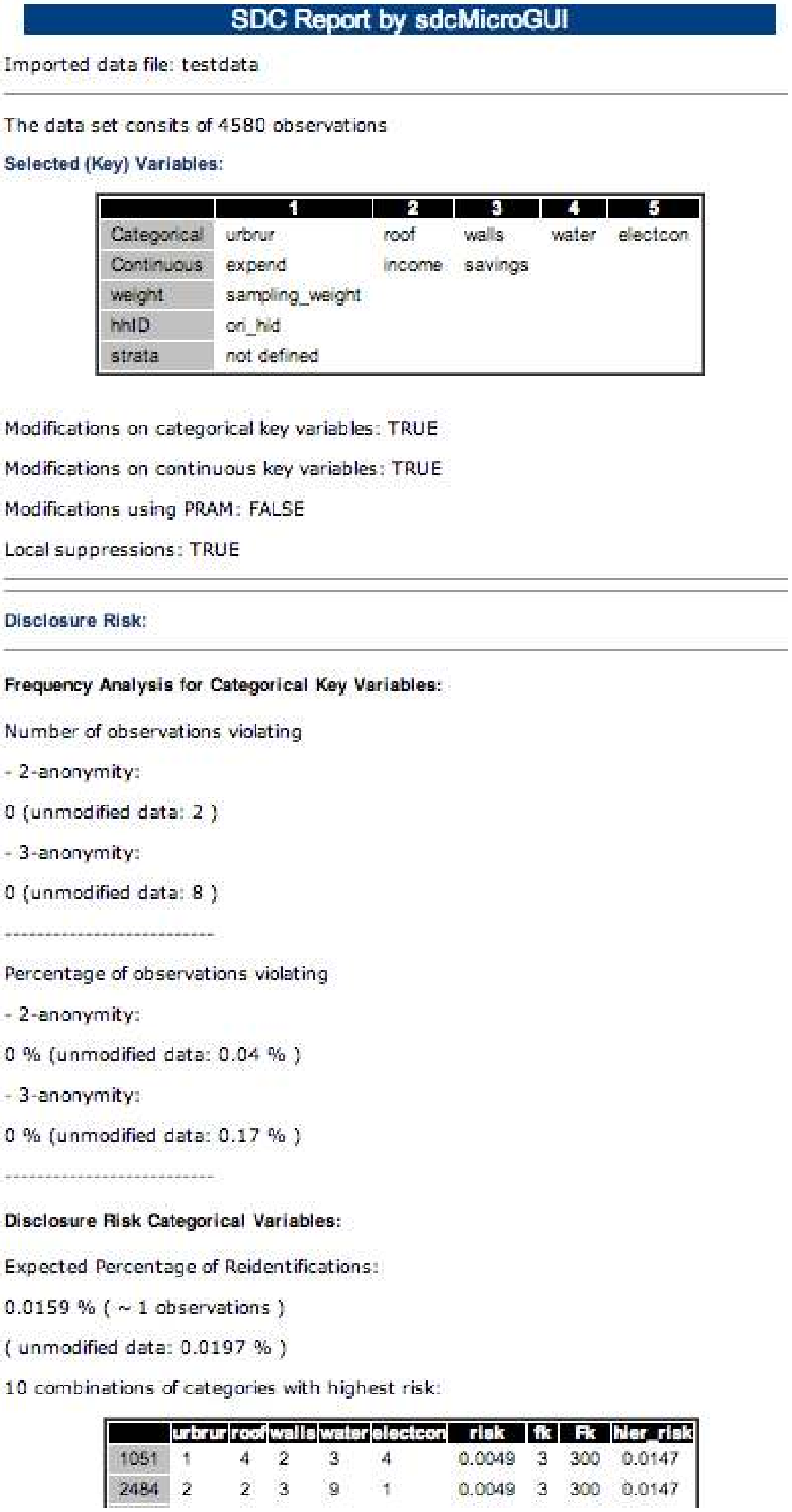
Report for Externals: this report includes less information. For example all information on disclosure risks, information loss is suppressed. This report is therefore suitable for external users of the anonymised data.

The first page of the internal report is shown in Figure 9. Information on the selected variables, the anonymisation methods applied as well as the disclosure risk is given at the first page. Detailed analysis on risk and utility follows in the report. The information that is included in the report always depends on the anonymisation process. For example, if PRAM was not applied, no specific summary for pramed variables is available. However, if PRAM has been used, the entire disclosure risk summary part is presented in differently.

6.3 The Script: Reproducibility of Results Obtained with the GUI

`sdcMicroGUI` provides reproducibility of any result obtained by clicking and setting parameters interactivly. This is one of the major features of the software because every action the user performed is internally stored, saved and listed and can be looked at in the *script frame*. To access this window, one needs to select *Script* → *View* on the main menu.

But it is not only possible to view the current script, but users can also export (*Script* → *Export*) and import (*Script* → *Import*) scripts from `sdcMicroGUI`.



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	hier_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.

Therefore, it is easily possible to reproduce previosly produced output. It is even suitable to modify some steps or alter the output. User can also remove specific steps from the script when navigating through it or execute only steps up to a certain point. This feature is very helpful to reproduce any results after some time,

Figure 10: The view script window showing the anonymization history

.

to be able to continue a previously started work and to restart some steps of the anonymization quickly.

7 Working with the sdcMicro Package

For each method explained we additionally show it's usage in software also via command line using the `sdcMicro` package. Therefore, a small introduction to the package is given before the implemented methods are explained.

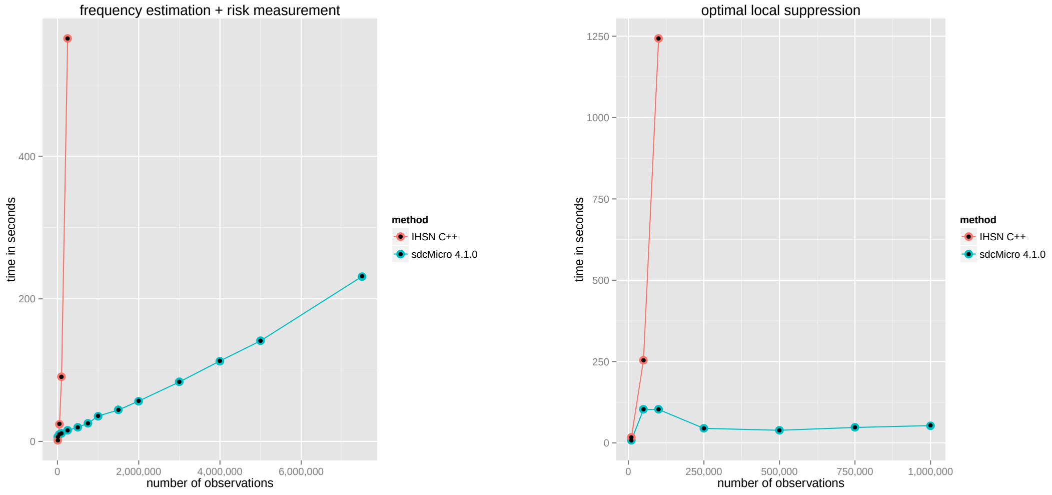
7.1 General Information about `sdcMicro`

In the last years, the statistical software environment `R` [R Development Core Team, 2011] (for short: `R`) gets more and more popular. Nowadays `R` has more users than any other statistical software[2], and `R` has got the standard statistical software for data analysis and graphics. For statisticians it has become the major programming language in its field.

The first version, version 1.0.0 of the `sdcMicro` package was realeased in 2007 on the comprehensive `R` archive network (CRAN, http://cran.r-project.org). The current release, version 4.1.0, is a huge step forward. Almost all methods are implemented in a highly object-oriented manner (using S4 classes) and they 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 which was partly integrated into `sdcMicro` and were partly rewritten. One example is given in Figure 11 where we show the computation time of the current version of `sdcMicro` ($\geq$ 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 where exponentional in computation time regarding the number of observations, the new implementation has linear complexity (see Figure 11(a)). For special tasks, e.g. for (heuristic optimal) local suppression, the computation time could even make faster, i.e. less than linear growth. The higher the number

[2] 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.

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 index of methods that are available in the package `sdcMicro` can be called by using the **help** function as shown in Listing 2. The package description shows a summary information about the package, see also Listing 1.

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

Listing 1: Accessing the index file to list the available methods in `sdcMicro`.

7.2 Getting Help

For each of the methods that has been implemented in `sdcMicro`, a help file is available. The help files not only describe all possible parameters that can be changed but that also feature simple, working examples that can directly be 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. An example is given in Listing 2.

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

Listing 2: Accessing the index of methods and the help file for function 'microaggregation' of `sdcMicro`.

`sdcMicro` features so called vignettes. These are manuals that are available in pdf-format. These vignettes contain interesting information, for example always the current version of these guidelines. Listing Listing 3 shows how to browse the available vignettes of `sdcMicro`.

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

Listing 3: Listing available vignettes of package `sdcMicro`.

7.3 S4 Class Structure

The `sdcMicro` package supports both, the straighforward application of methods to data and the application of methods to a so called *sdcMicroObj*. For example, when applying microaggregation on three continuous key variables on the data set 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 with, the `sdcMicro` package has to be loaded once in R as it is shown in Listing 4. This is however only possible, if the package was already installed on the computer. Installation instructions have already been given in section 1.2.

```
require(sdcMiro)
```

Listing 4: Loading the sdcMicro package

To define an object of class *sdcMicroObj*, the function **createSdcObj**() can to be used. In this case, all required parameters have to be specified. The neccessary parameters are for example categorical and continuous key variables, the vector of sampling weights and optionally stratification and cluster ID's. Listing 5 shows how to generate such an object using the test data that are 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')
```

Listing 5: Defining a sdcMicroObj.

We showed how to define the categorical and continous key variables, the vector of weights and the household IDs. The following slots of the *sdcMicroObj* sdc are pre-filled

```
> slotNames(sdc)

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

The first slot contains the original data, the second slot 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. 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 system already knows that and the user does not have to specify the variables again.

The application of a method to an *sdcMicroObj* is done by `method(sdcMicroObj)` ,

whereas `method` is a placeholder for any method available in `sdcMicro`. Listing 6 shows an example for this object-oriented implementation approach. In this example, the method microaggregation is applied on an object of class sdcMicroObj. Since microaggregation is only suitable for continuous scaled variables, the categorical variables remain untouched. Microaggregation is applied on all continuous key variables. Additionally, the risk and utility slots are updated and then contain the new estimates using current values of the microaggregated variables. In this example, default values for parameters are used, It is however possible, to change the default values. For details, see **help**(”microaggregation”).

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

Listing 6: Applying a method on an object of class sdcMicroObj.

The slots of the *sdcMicroObj* can be accessed also using function **get**.sdcMicroObj (), as it is shown in Listing 7.

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

Listing 7: Accessor to extract information from a sdcMicroObj.

Print methods are available to show the relevant information - see Listing ?? for printing the risk and the following result.

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

Listing 8: One - of many - print methods.

```
--------------------------
0 obs. with higher risk than the main part
Expected no. of re-identifications:
 4.33 [ 4.65 %]
--------------------------
--------------------------
Hierarchical risk
--------------------------
Expected no. 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. [2013].

References

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

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.

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

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. URL http://www.R-project.org. ISBN 3-900051-07-0.

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

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