Statistical Disclosure Control: A Practice Guide

Thijs Benschop, Cathrine Machingauta, Matthew Welch

Jul 11, 2018

TABLE OF CONTENT

CHAPTER

ONE

INTRODUCTION

National statistics agencies are mandated to collect microdata[1] from surveys and censuses to inform and measure policy effectiveness. In almost all countries, statistics acts and privacy laws govern these activities. These laws require that agencies protect the identity of respondents, but may also require that agencies disseminate the results and, in appropriate cases, the microdata. Data producers who are not part of national statistics agencies are also often subject to restrictions, through privacy laws or strict codes of conduct and ethics that require a similar commitment to privacy protection. This has to be balanced against the increasing requirement from funders that data produced using donor funds be made publically available.

This tension between complying with confidentiality requirements while at the same time requiring that microdata be released means that a demand exists for practical solutions for applying Statistical Disclosure Control (SDC), also known as microdata anonymization. The provision of adequate solutions and technical support has the potential to "unlock" a large number of datasets.

The International Household Survey Network (IHSN) and the World Bank have contributed to successful programs that have generated tools, resources and guidelines for the curation, preservation and dissemination of microdata and resulted in the documentation of thousands of surveys by countries and agencies across the world. While these programs have ensured substantial improvements in the preservation of data and dissemination of good quality metadata, many agencies are still reluctant to allow access to the microdata. The reasons are technical, legal, ethical and political, and sometimes involve a fear of being criticized for not providing perfect data. When combined with the tools and guidelines already developed by the IHSN/World Bank for the curation, preservation and dissemination of microdata, tools and guidelines for the anonymization of microdata should further reduce or remove some of these obstacles.

Working with the IHSN, PARIS21 (OECD), Statistics Austria and the Vienna University of Technology, the World Bank has contributed to the development of an open source software package for SDC, called *sdcMicro*. The package was developed for use with the open source *R* statistical software, available from the Comprehensive R Archive Network (CRAN) at http://cran.us.r-project.org. The package includes numerous methods for the assessment and reduction of disclosure risk in microdata.

Ensuring that a free open source solution is available to agencies was an important step forward, but not a sufficient one. There is still limited consolidated and reported knowledge on the impact of disclosure risk reduction methods on data utility. This limited access to knowledge combined with a lack of experience in using the tools and methods makes it difficult for many agencies to implement optimal solutions, i.e., solutions that meet their obligations towards both privacy protection and the release of data useful for policy monitoring and evaluation. This practice guide attempts to fill this critical gap by:

1. consolidating knowledge gained at the World Bank through experiments conducted during a large-scale evaluation of anonymization techniques
2. translating the experience and key results into practical guidelines

[1] Microdata are unit-level data obtained from sample surveys, censuses and administrative systems. They provide information about characteristics of individual people or entities such as households, business enterprises, facilities, farms or even geographical areas such as villages or towns. They allow in-depth understanding of socio-economic issues by studying relationships and interactions among phenomena. Microdata are thus key to designing projects and formulating policies, targeting interventions and monitoring and measuring the impact and results of projects, interventions and policies.

It should be stressed that SDC is only one part of the data release process, and its application must be considered within the complete data release framework. The level and methods of SDC depend on the laws of the country, the sensitivity of the data and the access policy (i.e., who will gain access) considered for release. Agencies that are currently releasing data are already using many of the methods described in this guide and applying appropriate access polices to their data before release. The primary objective of this guide is to provide a primer to those new to the process who are looking for guidance on both theory and practical implementation. This guide is not intended to prescribe or advocate for changes in methods that specific data producers are already using and which they have designed to fit and comply with their existing data release policies.

The guide seeks to provide practical steps to those agencies that want to unlock access to their data in a safe way and ensure that the data remain fit for purpose.

1.1 Building a knowledge base

The release of data is important, as it allows researchers and policymakers to replicate officially published results, generate new insights into issues, avoid duplication of surveys and provide greater returns to the investment in the survey process.

Both the production of reports, with aggregate tables of indicators and statistics, and the release of microdata result in privacy challenges to the producer. In the past, for many agencies, the only requirement was to release a report and some key indicators. The recent movement around Open Data, Open Government and transparency means that agencies are under greater pressure to release their microdata to allow broader use of data collected through public and donor funds. This guide focuses on the methods and processes for the release of microdata.

Releasing data in a safe way is required to protect the integrity of the statistical system, by ensuring agencies honor their commitment to respondents to protect their identity. Agencies do not widely share, in substantial detail, their knowledge and experience using SDC and the processes for creating safe data with other agencies. This makes it difficult for agencies new to the process to implement solutions. To fill this experience and knowledge gap, we evaluated the use of a broad suite of SDC methods on a range of survey microdata covering important development topics related to health, labor, education, poverty and inequality. The data we used were all previously treated to make them safe for release. Given that their producers had already treated these data, it was not possible, nor was it our goal, to pass any judgment on the safety of these data, many of which are in the public domain The focus was rather on measuring the effects that various methods would have on the risk-utility trade-off for microdata produced to measure common development indicators. We used the experience from this large-scale experimentation to inform our discussion of the processes and methods in this guide.

Important

At no point was any attempt made to re-identify, through matching or any other method, any respondents in the surveys we used in building our knowledge base. All risk assessments were based on frequencies and probabilities.

1.2 Using this guide

The methods discussed in this guide originate from a large body of literature on SDC. The processes underlying many of the methods are the subject of extensive academic research and many, if not all, of them are used extensively by agencies experienced in preparing microdata for release.

Where possible, for each method and topic, we provide elaborate examples, references to the original or seminal work describing the methods and algorithms in detail and recommended readings. This, when combined with the discussion of the method and practical considerations in this guide, should allow the reader to understand the methods and their strengths and weaknesses. It should also provide enough detail for readers to use an existing software solution to implement the methods or program the methods in statistical software of their choice.

For the examples in this guide, we use the open source and free package for SDC called *sdcMicro* as well as the statistical software *R*. *sdcMicro* is an add-on package to the statistical software *R*. The package was developed and is maintained by Matthias Templ, Alexander Kowarik and Bernhard Meindl.[2] The statistical software *R* and the *sdcMicro* package, as well as any other packages needed for the SDC process, are freely available from the Comprehensive R Archive Network (CRAN) mirrors (http://cran.r-project.org/). The software is available for Linux, Windows and Macintosh operating systems. We chose to use *R* and *sdcMicro* because it is freely available, accommodates all main data formats and is easy to adapt by the user. The World Bank, through the IHSN, has also provided funding towards the development of the *sdcMicro* package to ensure it meets the requirements of the agencies we support.

This guide does not provide a review of all other available packages for implementing the SDC process. Our concern is more with providing practical insight into the application of the methods. We would, however, like to highlight one particular other software package that is commonly used by agencies: μ-ARGUS[3]. μ-ARGUS is developed by Statistics Netherlands. *sdcMicro* and μ-ARGUS are both widely used in statistics offices in the European Union and implement many of the same methods.

The user needs some knowledge of *R* to use *sdcMicro*. It is beyond the scope of this guide to teach the use of *R*, but we do provide throughout the guide code examples on how to implement the necessary routines in *R*.[4] We also present a number of case studies that include the code for the anonymization of a number of demo datasets using *R*. Through these case studies, we demonstrate a number of approaches to the anonymization process in *R*.[5]

1.3 Outline of this guide

This guide is divided into the following main sections:

1. the Section Statistical Disclosure Control (SDC): An Introduction is a primer on SDC.
2. the Section Release Types gives an introduction to different release types for microdata.
3. the Sections Anonymization Methods , Measuring Risk and Measuring Utility and Information Loss cover SDC methods, risk and utility measurement. Here the goal is to provide knowledge that allows the reader to independently apply and execute the SDC process. This section is enriched with real examples as well as code snippets from the *sdcMicro* package. The interested reader can also find more information in the references and recommended readings at the end of each section.
4. the Section SDC with sdcMicro in R: Setting Up Your Data and more gives an overview of issues encountered when carrying out anonymization with the *sdcMicro* package in *R*, which exceed basic *R* knowledge. This section also includes tips and solutions to some of the common issues and problems that might be encountered when applying SDC methods in *R* with *sdcMicro*.
5. the Section The SDC Process provides a step-by-step guide to disclosure control, which draws upon the knowledge presented in the previous sections.
6. the Section Case Studies (Illustrating the SDC Process) presents a number of detailed case studies that demonstrate the use of the methods, their implementation in *sdcMicro* and the process that should be followed to reach the optimal risk-utility solution.

[2] See http://cran.r-project.org/web/packages/sdcMicro/index.html and the GitHub https://github.com/sdcTools/sdcMicro of the developers. The GitHub repository can also be used to submit bugs found in the package.

[3] μ-ARGUS is available at: http://neon.vb.cbs.nl/casc/mu.htm. The software was recently ported to open source.

[4] There are many free resources for learning *R* available on the web. One place to start would be the CRAN *R* Project page: http://cran.r-project.org/other-docs.html

[5] The developers of *sdcMicro* have also developed a graphical user interface (GUI) for the package, which is contained in the *sdcMicro* package available from the CRAN mirrors. The GUI, however, does not implement the full functionality of the *sdcMicro* package and is not discussed in this guide. The GUI can be called after loading sdcMicro by typing sdcApp() at the prompt.

CHAPTER

TWO

GLOSSARY AND LIST OF ACRONYMS

2.1 List of Acronyms

AFR Sub Saharan Africa

COICOP Classification of Individual Consumption by Purpose

CRAN Comprehensive R Archive Network

CTBIL Contingency Table-Based Information Loss

DHS Demographic and Health Surveys

DIS Data Intrusion Simulation

EAP East Asia and the Pacific

ECA Europe and Central Asia

EU European Union

GIS Geographical Information System

GPS Global Positioning System

GUI Graphical User Interface

HIV/AIDS Human Immunodeficiency Virus/Acquired Immune Deficiency Syndrome

I2D2 International Income Distribution Database

IHSN International Household Survey Network

LAC Latin America and the Caribbean

LSMS Living Standards Measurement Survey

MDAV Maximum Distance Average Vector

MDG Millennium Development Goal

MENA Middle East and North America

MICS Multiple Indicator Cluster Survey

MME Mean Monthly Expenditures

MMI Mean Monthly Income

MSU Minimal Sample Uniques

NSI National Statistical Institute

NSO National Statistical Office

OECD Organization for Economic Cooperation and Development

PARIS21 Partnership in Statistics for Development in the 21st century

PRAM Post Randomization Method

PC Principal Component

PUF Public Use File

SA South Asia

SDC Statistical Disclosure Control

SSE Sum of Squared Errors

SHIP Survey-based Harmonized Indicators Program

SUDA Special Uniques Detection Algorithm

SUF Scientific Use File

UNICEF United Nations Children's Fund

2.2 Glossary

Administrative data Data collected for administrative purposes by government agencies. Typically, administrative data require specific SDC methods.

Anonymization Use of techniques that convert confidential data into anonymized data/ removal or masking of identifying information from datasets.

Attribute disclosure Attribute disclosure occurs if an intruder is able to determine new characteristics of an individual or organization based on the information available in the released data.

Categorical variable A variable that takes values over a finite set, e.g., gender. Also called factor in *R*.

Confidentiality Data confidentiality is a property of data, usually resulting from legislative measures, which prevents it from unauthorized disclosure.[2]

Confidential data Data that will allow identification of an individual or organization, either directly or indirectly.[1]

Continuous variable A variable with which numerical and arithmetic operations can be performed, e.g., income.

Data protection Data protection refers to the set of privacy-motivated laws, policies and procedures that aim to minimize intrusion into respondents' privacy caused by the collection, storage and dissemination of personal data.[2]

Deterministic methods Anonymization methods that follow a certain algorithm and produce the same results if applied repeatedly to the same data with the same set of parameters.

Direct identifier A variable that reveals directly and unambiguously the identity of a respondent, e.g., names, social identity numbers.

Disclosure Disclosure occurs when a person or an organization recognizes or learns something that they did not already know about another person or organization through released data.[1] See also Identity disclosure, Attribute disclosure and Inferential disclosure.

Disclosure risk A disclosure risk occurs if an unacceptably narrow estimation of a respondent's confidential information is possible or if exact disclosure is possible with a high level of confidence.[2] Disclosure risk also refers to the probability that successful disclosure could occur.

[2] OECD, http://stats.oecd.org/glossary

[1] Australian Bureau of Statistics, http://www.nss.gov.au/nss/home.nsf/pages/Confidentiality+-+Glossary

End user The user of the released microdata file after anonymization. Who is the end user depends on the release type.

Factor variable Factor variables are one way to classify categorical variables in *R*.

Hierarchical structure Data is made up of collections of records that are interconnected through links, e.g., individuals belonging to groups/households or employees belonging to companies.

Identifier An identifier is a variable/ information that can be used to establish identity of an individual or organization. Identifiers can lead to direct or indirect identification.

Identity disclosure Identity disclosure occurs if an intruder associates a known individual or organization with a released data record.

Indirect identification Indirect identification occurs when the identity of an individual or organization is disclosed, not using direct identifiers but through a combination of unique characteristics in key variables.[1]

Inferential disclosure Inferential disclosure occurs if an intruder is able to determine the value of some characteristic of an individual or organization more accurately with the released data than otherwise would have been possible.

Information loss Information loss refers to the reduction of the information content in the released data relative to the information content in the raw data. Information loss is often measured with respect to common analytical measures, such as regressions and indicators. See also Utility.

Interval A set of numbers between two designated endpoints that may or may not be included. Brackets (e.g., [0, 1]) denote a closed interval, which includes the endpoints 0 and 1. Parentheses (e.g., (0, 1) denote an open interval, which does not include the endpoints.

Intruder A user who misuses released data by trying to disclose information about an individual or organization, using a set of characteristics known to the user.

k-anonymity The risk measure k-anonymity is based on the principle that the number of individuals in a sample sharing the same combination of values (key) of categorical key variables should be higher than a specified threshold k.

Key A combination or pattern of key variables/quasi-identifiers.

Key variables A set of variables that, in combination, can be linked to external information to re-identify respondents in the released dataset. Key variables are also called "quasi-identifiers" or "implicit identifiers".

Microaggregation Anonymization method that is based on replacing values for a certain variable with a common value for a group of records. The grouping of records is based on a proximity measure of variables of interest. The groups of records are also used to calculate the replacement value.

Microdata A set of records containing information on individual respondents or on economic entities. Such records may contain responses to a survey questionnaire or administrative forms.

Noise addition Anonymization method based on adding or multiplying a stochastic or randomized number to the original values to protect data from exact matching with external files. Noise addition is typically applied to continuous variables.

Non-perturbative methods Anonymization methods that reduce the detail in the data or suppress certain values (masking) without distorting the data structure.

Observation A set of data derived from an object/unit of experiment, e.g., an individual (in individual-level data), a household (in household-level data) or a company (in company data). Observations are also called "records".

Original data The data before SDC/anonymization methods were applied. Also called "raw data" or "untreated data".

Outlier An unusual value that is correctly reported but is not typical of the rest of the population. Outliers can also be observations with an unusual combination of values for variables, such as 20-year-old widow. On their own age, 20 and widow are not unusual values, but their combination may be.[1]

Perturbative methods Anonymization methods that alter values slightly to limit disclosure risk by creating uncertainty around the true values, while retaining as much content and structure as possible, e.g. microaggregation and noise addition.

Population unique The only record in the population with a particular set of characteristics, such that the individual or organization can be distinguished from other units in the population based on that set of characteristics.

Post Randomization Method (PRAM) Anonymization method for microdata in which the scores of a categorical variable are altered according to certain probabilities. It is thus intentional misclassification with known misclassification probabilities.[1]

Probabilistic methods Anonymization methods that depend on a probability mechanism or a random number-generating mechanism. Every time a probabilistic method is used, a different outcome is generated.

Privacy Privacy is a concept that applies to data subjects while confidentiality applies to data. The concept is defined as follows: "It is the status accorded to data which has been agreed upon between the person or organization furnishing the data and the organization receiving it and which describes the degree of protection which will be provided."[2]

Public Use File (PUF) Type of release of microdata file, which is freely available to any user, for example on the internet.

Quasi-identifiers A set of variables that, in combination, can be linked to external information to re-identify respondents in the released dataset. Quasi-identifiers are also called "key variables" or "implicit identifiers".

Raw data The data before SDC/anonymization methods were applied. Also called "original data" or "untreated data".

Recoding Anonymization method for microdata in which groups of existing categories/values are replaced with new values, e.g. the values 'protestant', and 'catholic' are replaced with 'Christian'. Recoding reduces the detail in the data. Recoding of continuous variables leads to a transformation from continuous to categorical, e.g. creating income bands.

Record A set of data derived from an object/unit of experiment, e.g., an individual (in individual-level data), a household (in household-level data) or a company (in company data). Records are also called "observations".

Regression A statistical process of measuring the relation between the mean value of one variable and corresponding values of other variables.

Re-identification risk See Disclosure risk

Release Dissemination – the release to users of information obtained through a statistical activity.[2]

Respondents Individuals or units of observation whose information/responses to a survey make up the data file.

Sample unique The only record in the sample with a particular set of characteristics, such that the individual or organization can be distinguished from other units in the sample based on that set of characteristics.

Scientific Use File (SUF) Type of release of microdata file, which is only available to selected researchers under contract. Also known as "licensed file", "microdata under contract" or "research file".

sdcMicro An *R* based package authored by Templ, M., Kowarik, A. and Meindl, B. with tools for the anonymization of microdata, i.e. for the creation of public- and scientific-use files.

sdcMicroGUI A GUI for the *R* based *sdcMicro* package, which allows users to use the *sdcMicro* tools without *R* knowledge.

Sensitive variables Sensitive or confidential variables are those whose values must not be discovered for any respondent in the dataset. The determination of sensitive variables is often subject to legal and ethical concerns.

Statistical Disclosure Control (SDC) Statistical Disclosure Control techniques can be defined as the set of methods to reduce the risk of disclosing information on individuals, businesses or other organizations. Such methods are only related to the dissemination step and are usually based on restricting the amount of or modifying the data released.[2]

Suppression Data suppression involves not releasing information that is considered unsafe because it fails confidentiality rules being applied. Sometimes this is done is by replacing values signifying individual attributes with missing values. In the context of this guide, usually to achieve a desired level of *k*- anonymity.

Threshold An established level, value, margin or point at which values that fall above or below it will deem the data safe or unsafe. If unsafe, further action will need to be taken to reduce the risk of identification.

Utility Data utility describes the value of data as an analytical resource, comprising analytical completeness and analytical validity.

Untreated data The data before SDC/anonymization methods were applied. Also called "raw data" or "original data".

Variable Any characteristic, number or quantity that can be measured or counted for each unit of observation.

CHAPTER

THREE

STATISTICAL DISCLOSURE CONTROL (SDC): AN INTRODUCTION

3.1 Need for SDC

A large part of the data collected by statistical agencies cannot be published directly due to privacy and confidentiality concerns. These concerns are both of legal and ethical nature. SDC seeks to treat and alter the data so that the data can be published or released without revealing the confidential information it contains, while, at the same time, limit information loss due to the anonymization of the data. In this guide, we discuss only disclosure control for microdata.[1] Microdata are datasets that provide information on a set of variables for each individual respondent. Respondents can be natural persons, but also legal entities such as companies.

The aim of anonymizing microdata is to transform the datasets to achieve an "acceptable level" of disclosure risk. The level of acceptability of disclosure risk and the need for anonymization are usually at the discretion of the data producer and guided by legislation. These are formulated in the dissemination policies and programs of the data providers and based on considerations including "[. . .] the costs and expertise involved; questions of data quality, potential misuse and misunderstanding of data by users; legal and ethical matters; and maintaining the trust and support of respondents" (*DuBo10*). There is a moral, ethical and legal obligation for the data producers to ensure that data provided by the respondents are used only for statistical purposes.

In some cases, the dissemination of microdata is a legal obligation, but, in most cases, the legislation will formulate restrictions. Thus, a country's legislative framework will shape its microdata dissemination policy. It is crucial for data producers to "ensure there is a sound legal and ethical base (as well as the technical and methodological tools) for protecting confidentiality. This legal and ethical base requires a balanced assessment between the public good of confidentiality protection on the one hand, and the public benefits of research on the other. A decision on whether or not to provide access might depend on the merits of specific research proposals and the credibility of the researcher, and there should be some allowance for this in the legal arrangements." (*DuBo10*).

"Data access arrangements should respect the legal rights and legitimate interests of all stakeholders in the public research enterprise. Access to, and use of, certain research data will necessarily be limited by various types of legal requirements, which may include restrictions for reasons of:

- National security: data pertaining to intelligence, military activities, or political decision making may be classified and therefore subject to restricted access.
- Privacy and confidentiality: data on human subjects and other personal data are subject to restricted access under national laws and policies to protect confidentiality and privacy. However, anonymization or confidentiality procedures that ensure a satisfactory level of confidentiality should be considered by custodians of such data to preserve as much data utility as possible for researchers.
- Trade secrets and intellectual property rights: data on, or from, businesses or other parties that contain confidential information may not be accessible for research. (. . .)" (*DuBo10*).

Box 1, extracted from *DuBo10*, provides several examples of statistical legislation on microdata release.

[1] There is another strand of literature on the anonymization of tabular data, see e.g., *HDFG12*.

Info-box - Examples of statistical legislation on microdata release

1. The US Bureau of the Census operates under Title 13-Census of the US Code. "Title 13, U.S.C., Section 9 prohibits the publication or release of any information that would permit identification of any particular establishment, individual, or household. Disclosure assurance involves the steps taken to ensure that Title 13 data prepared for public release will not result in wrongful disclosure. This includes both the use of disclosure limitation methods and the review process to ensure that the disclosure limitation techniques used provide adequate protection to the information."

Source: https://www.census.gov/srd/sdc/wendy.drb.faq.pdf

2. In Canada the Statistics Act states that "no person who has been sworn under section 6 shall disclose or knowingly cause to be disclosed, by any means, any information obtained under this Act in such a manner that it is possible from the disclosure to relate the particulars obtained from any individual return to any identifiable individual person, business or organization."

Source: http://laws-lois.justice.gc.ca/eng/acts/S-19/page-2.html

Statistics Canada does release microdata files. Its microdata release policy states that Statistics Canada will authorise the release of microdata files for public use when:

1. the release substantially enhances the analytical value of the data collected; and
2. the Agency is satisfied all reasonable steps have been taken to prevent the identification of particular survey units.
3. In Thailand the Act states: "Personal information obtained under this act shall be strictly considered confidential. A person who performs his or her duty hereunder or a person who has the duty of maintaining such information cannot disclose it to anyone who doesn't have a duty hereunder except in the case that:
   (a) Such disclosure is for the purpose of any investigation or legal proceedings in a case relating to an offense hereunder.
   (b) Such disclosure is for the use of agencies in the preparation, analysis or research of statistics provided that such disclosure does not cause damage to the information owner and does not identify or disclose the data owner."

Source: http://web.nso.go.th/eng/en/about/stat_act2007.pdf section 15.

Besides the legal and ethical concerns and codes of conducts of agencies producing statistics, SDC is important because it guarantees data quality and response rates in future surveys. If respondents feel that data producers are not protecting their privacy, they might not be willing to participate in future surveys. "[...] one incident, particularly if it receives strong media attention, could have a significant impact on respondent cooperation and therefore on the quality of official statistics" (*DuBo10*). At the same time, if data users are unable to gain enough utility from the data due to excessive or inappropriate SDC protection, or are unable to access the data, then the large investment in producing the data will be lost.

3.2 The risk-utility trade-off in the SDC process

SDC is characterized by the trade-off between risk of disclosure and utility of the data for end users. The risk–utility scale extends between two extremes; (i) no data is released (zero risk of disclosure) and thus users gain no utility from the data, to (ii) data is released without any treatment, and thus with maximum risk of disclosure, but also maximum utility to the user (i.e., no information loss). The goal of a well-implemented SDC process is to find the optimal point where utility for end users is maximized at an acceptable level of risk. Fig. 3.1 illustrates this trade-off. The triangle corresponds to the raw data. The raw data have no information loss, but generally have a disclosure risk higher than the acceptable level. The other extreme is the square, which corresponds to no data release. In that case there is no

disclosure risk, but also no utility from the data for the users. The points in-between correspond to different choices of SDC methods and/or parameters for these methods applied to different variables. The SDC process looks for the SDC methods and the parameters for those methods and applies these in a way that reduces the risk sufficiently, while minimizing the information loss.

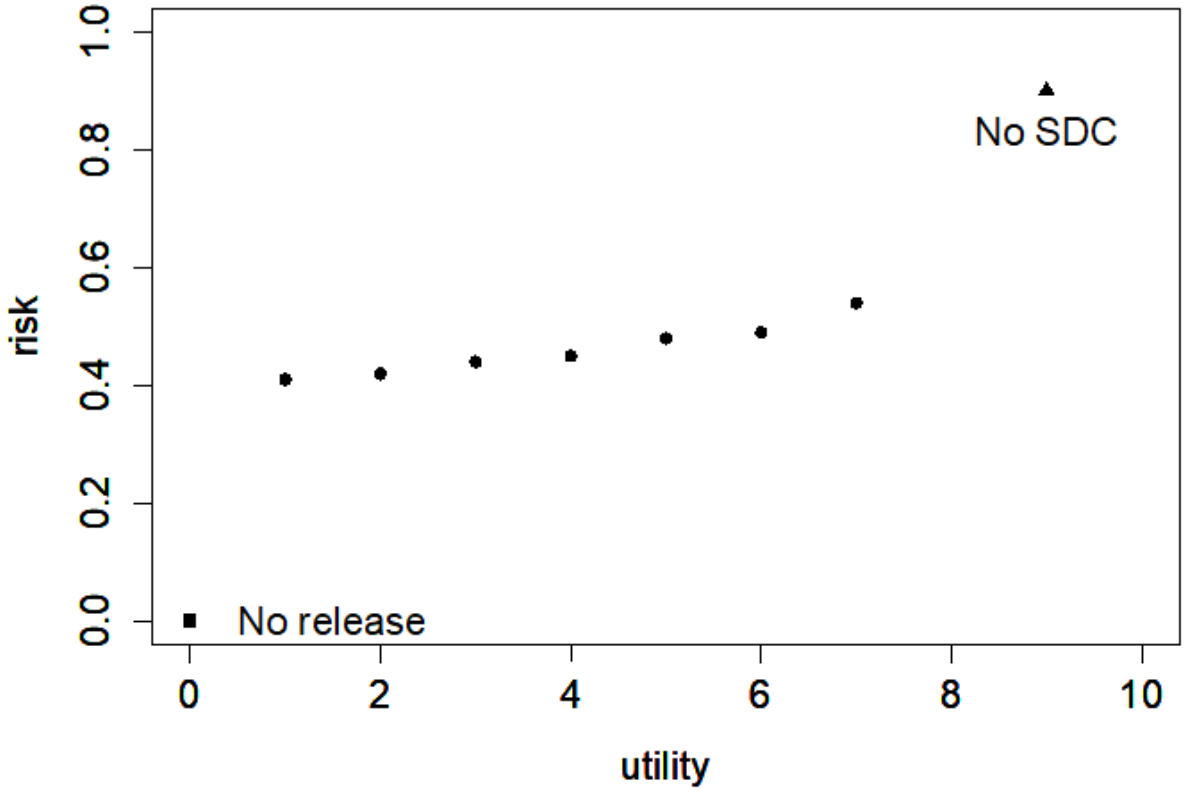

Fig. 3.1: Risk-utility trade-off

SDC cannot achieve total risk elimination, but can reduce the risk to an acceptable level. Any application of SDC methods will suppress or alter values in the data and as such decrease the utility (i.e., result in information loss) when compared to the original data. A common thread that will be emphasized throughout this guide will be that the process of SDC should prioritize the goal of protecting respondents, while at the same time keeping the data users in mind to limit information loss. In general, the lower the disclosure risk, the higher the information loss and the lower the data utility for end-users.

In practice, choosing SDC methods is partially trial and error: after applying methods, disclosure risk and data utility are re-measured and compared to the results of other choices of methods and parameters. If the result is satisfactory, the data can be released. We will see that often the first attempt will not be the optimal one. The risk may not be sufficiently reduced or the information loss may be too high and the process has to be repeated with different methods or parameters until a satisfactory solution is found. Disclosure risk, data utility and information loss in the SDC context and how to measure them are discussed in subsequent chapters of this guide.

Again, it must be stressed that the level of SDC and methods applied depend to a large extent on the entire data release framework. For example, a key consideration is to whom and under what conditions the data are to be released (see also the Section Release types). If data are to be released as public use data, then the level of SDC applied will necessarily need to be higher than in the cases where data are released under license conditions to trusted users after careful vetting. With careful preparation, data may be released under both public and licensed versions. We discuss

how this might be achieved later in the guide.

References

CHAPTER

FOUR

RELEASE TYPES

This section discusses data release. Rather than rewriting work that has already been conducted through the World Bank and its partners at the IHSN, this section extracts from an excellent guide published by *DuBo10*.

The trade-off between risk and utility in the anonymization process depends greatly on who the users are[1] and under what conditions a microdata file is released. Generally, three types of data release methods are practiced and apply to different target groups.

- **Public Use File (PUF)**: the data "are available to anyone agreeing to respect a core set of easy-to-meet conditions. Such conditions relate to what cannot be done with the data (e.g. the data cannot be sold), upon gaining access to the data. In some cases PUFs are disseminated with no conditions; often being made available on-line, [e.g. on the website of the statistical agency]. These data are made easily accessible because the risk of identifying individual respondents is considered minimal. Minimising the risk of disclosure involves eliminating all content that can identify respondents directly—for instance, names, addresses and telephone numbers. In addition this requires purging relevant indirect identifiers from the microdata file. These vary across survey designs, but commonly-suppressed indirect identifiers include geographical information below the sub-national level at which the sample is representative. Occasionally, certain records may be suppressed also from PUFs, as might variables characterised by extremely skewed distribution or outliers. However, in lieu of deleting entire records or variables from microdata files, alternative SDC methods can minimise the risk of disclosure while maximizing information content. Such methods include top-and-bottom coding, local suppression or using data perturbation techniques [(see the Section Anonymization methods for an overview of anonymization methods)]. PUFs are typically generated from census data files using a sub-set [or sample] of records rather than the entire file and [from sample surveys, such as] household surveys." (*DuBo10*).
- **Scientific Use File (SUF)** (also known as a licensed file, microdata under contract or research file): the "dissemination is restricted to users who have received authorization to access them after submitting a documented application and signing an agreement governing the data's use. While typically licensed files are also anonymised to ensure the risk of identifying individuals is minimised when used in isolation, they may still [potentially] contain identifiable data if linked with other data files. Direct identifiers such as respondents' names must be removed from a licensed dataset. The data files may, however, still contain indirect variables that could identify respondents by matching them to other data files such as voter lists, land registers or school records. When disseminating licensed files, the recommendation is to establish and sign an agreement between the data producer and external bona fide users – trustworthy users with legitimate need to access the data. Such an agreement should govern access and use of such microdata files[2]. Sometimes, licensing agreements are only entered into with users affiliated to an appropriate sponsoring institution. i.e., research centers, universities or development partners. It is further recommended that, before entering into a data access and use agreement, the data producer asks potential users to complete an application form to demonstrate the need to use a licensed file (instead of the PUF version, if available) for a stated statistical or research purpose" (*DuBo10*). This also allows the data producer to learn which characteristics of the data are important for the users, which is valuable information for optimizing future anonymization processes.

[1] See Section 5 in *DuBo10* as to who the users of microdata are and to whom microdata should be made available.
[2] Appendix B provides an example of a blanket agreement.

- **Microdata available in a controlled research data center** (also known as data enclave): "Some files may be offered to users under strict conditions in a data enclave. This is a facility [(often on the premises of the data provider)] equipped with computers not linked to the internet or an external network and from which no information can be downloaded via USB ports, CD-DVD or other drives. Data enclaves contain data that are particularly sensitive or allow direct or easy identification of respondents. Examples include complete population census datasets, enterprise surveys and certain health related datasets containing highly-confidential information. Users interested in accessing a data enclave will not necessarily have access to the full dataset – only to the particular data subset they require. They will be asked to complete an application form demonstrating a legitimate need to access these data to fulfill a stated statistical or research purpose [...] The outputs generated must be scrutinised by way of a full disclosure review before release. Operating a data enclave may be expensive – it requires special premises and computer equipment. It also demands staff with the skills and time to review outputs before their removal from the data enclave in order to ensure there is no risk of disclosure. Such staff must be familiar with data analysis and be able to review the request process and manage file servers. Because of the substantial operating costs and technical skills required, some statistical agencies or other official data producers opt to collaborate with academic institutions or research centres to establish and manage data enclaves."

There are other data access possibilities besides these, such as teaching files, files for other specific purposes, remote execution or remote access. Obviously, the required level of protection depends on the type of release; a PUF file must be protected to a much larger extent than a SUF file, which in turn has to be protected more than a file which is only available in an on-site facility. The Section Step 3: Type of release gives more guidance on the choice of the release type and its implications for the anonymization process. The same microdata set can be released in different ways for different users, e.g., as SUF and teaching file. The Section Step 3: Type of release discusses the particular issues of multiple releases of one dataset.

The first step for any agency that wants to release data would be formulation of clear data dissemination policies for the release of microdata. We will see later that deciding on the level of anonymization needed will depend partly on knowing under what conditions the data will be released. Access policies and conditions provide the framework for the whole release process.

The following sections further specify the conditions under which microdata should be provided under different release types.

4.1 Conditions for PUFs

"Generally, data regarded as public are open to anyone with access to an [National Statistical Office] (NSO) website. It is, however, normally good practice to include statements defining suitable uses for and precautions to be adopted in using the data. While these may not be legally binding, they serve to sensitise the user. Prohibitions such as attempts to link the data to other sources can be part of the 'use statement' to which the user must agree, on-line, before the data can be downloaded. [...] Dissemination of microdata files necessarily involves the application of rules or principles. [The info-box] below [taken from *DuBo10*] shows basic principles normally applying to PUFs." (*DuBo10*).

Info-box - Conditions for accessing and using PUFs

1. Data and other material provided by the NSO will not be redistributed or sold to other individuals, institutions or organisations without the NSO's written agreement.
2. Data will be used for statistical and scientific research purposes only. They will be employed solely for reporting aggregated information, including modelling, and not for investigating specific individuals or organisations.
3. No attempt will be made to re-identify respondents, and there will be no use of the identity of any person or establishment discovered inadvertently. Any such discovery will be reported immediately to the NSO.
4. No attempt will be made to produce links between datasets provided by the NSO or between NSO data and other datasets that could identify individuals or organisations.

5. Any books, articles, conference papers, theses, dissertations, reports or other publications employing data obtained from the NSO will cite the source, in line with the citation requirement provided with the dataset.
6. An electronic copy of all publications based on the requested data will be sent to the NSO.
7. The original collector of the data, the NSO, and the relevant funding agencies bear no responsibility for the data's use or interpretation or inferences based upon it.

Note: Items 3 and 6 in the list require that users be provided with an easy way to communicate with the data provider. It is good practice to provide a contact number, an email address, and possibly an on-line "feedback provision" system.

Source: *DuBo10*

4.2 Conditions for SUFs

"For [SUFs], terms and conditions must include the basic common principles plus some additional ones applying to the researcher's organisation. There are two options: firstly, data are provided to a researcher or a team for a specific purpose; secondly, data are provided to an organization under a blanket agreement for internal use, e.g., to an international body or research agency. In both cases, the researcher's organisation must be identified, as must suitable representatives to sign the licence" (*DuBo10*).

Access to a researcher or research team for a specific purpose

"If data are provided for an individual research project, the research team must be identified. This is covered by requiring interested users to complete a formal request to access the data (a model of such a request form is provided in Appendix 1 [in *DuBo10*]). The conditions to obtain the data (see example in the info-box below) will specify that the files will not be shared outside the organisation and that data will be stored securely. To the possible extent, the intended use of the data – including a list of expected outputs and the organisation's dissemination policy – must be identified. Access to licensed datasets is only granted when there is a legally-registered sponsoring agency, e.g., government ministry, university, research centre or national or international organization" (*DuBo10*).

Info-box - Conditions for accessing and using SUFs

Note: Items 1 to 8 below are similar to the conditions for use of public use files in the info-box above. Items 9 and 10 would have to be adapted in the case of a blanket agreement.

1. Data and other material provided by the NSO will not be redistributed or sold to other individuals, institutions or organisations without the NSO's written agreement.
2. Data will be used for statistical and scientific research purposes only. They will be employed solely for reporting aggregated information, including modelling, and not for investigating specific individuals or organisations.
3. No attempt will be made to re-identify respondents, and there will be no use of the identity of any person or establishment discovered inadvertently. Any such discovery will be reported immediately to the NSO.
4. No attempt will be made to produce links between datasets provided by the NSO or between NSO data and other datasets that could identify individuals or organisations.
5. Any books, articles, conference papers, theses, dissertations, reports or other publications employing data obtained from the NSO will cite the source, in line with the citation requirement provided with the dataset.
6. An electronic copy of all publications based on therequested data will be sent to the NSO.
7. The NSO and the relevant funding agencies bear no responsibility the data's use or for interpretation or inferences based upon it.
8. An electronic copy of all publications based on the requested data will be sent to the NSO.

9. The researcher's organisation must be identified, as must the principal and other researchers involved in using the data must be identified. The principal researcher must sign the licence on behalf of the organization. If the principal researcher is not authorized to sign on behalf of the receiving organization, a suitable representative must be identified.
10. The intended use of the data, including a list of expected outputs and the organisation's dissemination policy must be identified.

(Conditions 9 to 11 may be waved in the case of educational institutions)

Source: *DuBo10*

Blanket agreement to an organization

"In the case of a blanket agreement, where it is agreed the data can be used widely but securely within the receiving organisation, the licence should ensure compliance, with a named individual formally assuming responsibility for this. Each additional user must be made aware of the terms and conditions that apply to data files: this can be achieved by having to sign an affidavit. Where such an agreement exists, with security in place, it is not necessary for users to destroy the data after use" (*DuBo10*). Appendix B provides an example of the formulation of such an agreement.

4.3 Conditions for microdata available in a controlled research data center

Access to microdata in research data centers is "used for particularly sensitive data or for more detailed data for which sufficient anonymisation to release them outside the NSO premises is not possible. These can be referred to also as data laboratories or research data centres. A [research data centre] may be located at the NSO headquarters or in major centres such as universities close to the research community. They are used to give researchers access to complete data files but without the risk of releasing confidential data. In a typical [research data centre], NSO staff supervise access and use of the data; the computers must not be able to communicate outside the [research data centre]; and the results obtained by the researchers must be screened for confidentiality by an NSO analyst before taken outside. A model of a data enclave access policy is provided in Appendix 2 [in *DuBo10*], and a model of a data enclave access request form is in Appendix 3 [in *DuBo10*]" (*DuBo10*).

Research data centers "have the advantage of providing access to detailed microdata but the disadvantage of requiring researchers to work at a different location. And they are expensive to set up and operate. It is, however, quite likely that many countries have used on-site researchers as a way of providing access to microdata. These researchers are sworn in under the statistics' acts in the same way as regular NSO employees. This approach tends to favour researchers who live near NSO headquarters." (*DuBo10*)

Recommended Reading Material on Release Types

Dupriez, O., & Boyko, E. (2010). *Dissemination of Microdata Files; Principles, Procedures and Practices.* International Household Survey Network (IHSN).

References

CHAPTER

FIVE

MEASURING RISK

5.1 Types of disclosure

Measuring disclosure risk is an important part of the SDC process: risk measures are used to judge whether a data file is safe enough for release. Before measuring disclosure risk, we have to define what type of disclosure is relevant for the data at hand. The literature commonly defines three types of disclosure; we take these directly from *Lamb93* (see also *HDFG12*).

- **Identity disclosure**, which occurs if the intruder associates a known individual with a released data record. For example, the intruder links a released data record with external information, or identifies a respondent with extreme data values. In this case, an intruder can exploit a small subset of variables to make the linkage, and once the linkage is successful, the intruder has access to all other information in the released data related to the specific respondent.
- **Attribute disclosure**, which occurs if the intruder is able to determine some new characteristics of an individual based on the information available in the released data. Attribute disclosure occurs if a respondent is correctly re-identified and the dataset contains variables containing information that was previously unknown to the intruder. Attribute disclosure can also occur without identity disclosure. For example, if a hospital publishes data showing that all female patients aged 56 to 60 have cancer, an intruder then knows the medical condition of any female patient aged 56 to 60 in the dataset without having to identify the specific individual.
- **Inferential disclosure**, which occurs if the intruder is able to determine the value of some characteristic of an individual more accurately with the released data than would otherwise have been possible. For example, with a highly predictive regression model, an intruder may be able to infer a respondent's sensitive income information using attributes recorded in the data, leading to inferential disclosure.

SDC methods for microdata are intended to prevent identity and attribute disclosure. Inferential disclosure is generally not addressed in SDC in the microdata setting, since microdata is distributed precisely so that researchers can make statistical inference and understand relationships between variables. In that sense, inference cannot be likened to disclosure. Also, inferences are designed to predict aggregate, not individual, behavior, and are therefore usually poor predictors of individual data values.

5.2 Classification of variables

For the purpose of the SDC process, we use the classifications of variables described in the following paragraphs (see Fig. 5.1 for an overview). The initial classification of variables into identifying and non-identifying variables depends on the way the variables can be used by intruders for re-identification (*HDFG12*, *TeMK14*):

- **Identifying variables:** these contain information that can lead to the identification of respondents and can be further categorized as:

  - **Direct identifiers** reveal directly and unambiguously the identity of the respondent. Examples are names, passport numbers, social identity numbers and addresses. Direct identifiers should be removed from the dataset prior to release. Removal of direct identifiers is a straightforward process and always the first step in producing a safe microdata set for release. Removal of direct identifiers, however, is often not sufficient.
  - **Quasi-identifiers** (**or key variables**) contain information that, when combined with other quasi-identifiers in the dataset, can lead to re-identification of respondents. This is especially the case when they can be used to match the information with other external information or data. Examples of quasi-identifiers are race, birth date, sex and ZIP/postal codes, which might be easily combined or linked to publically available external information and make identification possible. The combinations of values of several quasi-identifiers are called keys (see also the section *Levels of Risk*). The values of quasi-identifiers themselves often do not lead to identification (e.g., male/female), but a combination of several values of quasi-identifier can render a record unique (e.g. male, 14 years, married) and hence identifiable. It is not generally advisable to simply remove quasi-identifiers from the data to solve the problem. In many cases, they will be important variables for any sensible analysis. In practice, any variable in the dataset could potentially be used as a quasi-identifier. SDC addresses this by identifying variables as quasi-identifiers and anonymizing them while still maintaining the information in the dataset for release.

- **Non-identifying** variables are variables that cannot be used for re-identification of respondents. This could be because these variables are not contained in any other data files or other external sources and are not observable to an intruder. Non-identifying variables are nevertheless important in the SDC process, since they may contain confidential/sensitive information, which may prove damaging should disclosure occur as a result of identity disclosure based on identifying variables.

These classifications of variables depend partially on the availability of external datasets that might contain information that, when combined with the current data, could lead to disclosure. The identification and classification of variables as quasi-identifiers depends, amongst others, on the availability of information in external datasets. An important step in the SDC process is to define a list of possible disclosure scenarios based on how the quasi-identifiers might be combined with each other and information in external datasets and then treating the data to prevent disclosure. We discuss disclosure scenarios in more detail in the section *Disclosure scenarios*.

For the SDC process, it is also useful to further classify the quasi-identifiers into **categorical**, **continuous** and **semi-continuous** variables. This classification is important for determining the appropriate SDC methods for that variable, as well as the validity of risk measures.

- **Categorical** variables take values over a finite set, and any arithmetic operations using them are generally not meaningful or not allowed. Examples of categorical variables are gender, region and education level.
- **Continuous** variables can take on an infinite number of values in a given set. Examples are income, body height and size of land plot. Continuous variables can be transformed into categorical variables by constructing intervals (such as income bands).[1]
- **Semi-continuous** variables are continuous variables that take on values that are limited to a finite set. An example is age measured in years, which could take on values in the set $\{0, 1, \ldots, 100\}$. The finite nature of the values for these variables means that they can be treated as categorical variables for the purpose of SDC.[2]

Apart from these classifications of variables, the SDC process further classifies variables according to their sensitivity or confidentiality. Both quasi-identifiers and non-identifying variables can be classified as **sensitive** (or confidential) or **non-sensitive** (or non-confidential). This distinction is not important for direct identifiers, since direct identifiers are removed from the released data.

- **Sensitive** variables contain confidential information that should not be disclosed without suitable treatment using SDC methods to reduce disclosure risk. Examples are income, religion, political affiliation and variables

[1] Recoding a continuous variable is sometimes useful in cases where the data contains only a few continuous variables. We will see in the Section *Individual risk* that many methods used for risk calculation depend on whether the variables are categorical. We will also see that it is easier for the measurement of risk if the data contains only categorical or only continuous variables.

[2] This is discussed in greater detail in the following sections. In cases where the number of possible values is large, recoding the variable, or parts of the set it takes values on, to obtain fewer distinct values is recommended.

concerning health. Whether a variable is sensitive depends on the context and country: a certain variable can be considered sensitive in one country and non-sensitive in another.

- **Non-sensitive** variables contain non-confidential information on the respondent, such as place of residence or rural/urban residence. The classification of a variable as non-sensitive, however, does not mean that it does not need to be considered in the SDC process. Non-sensitive variables may still serve as quasi-identifiers when combined with other variables or other external data.

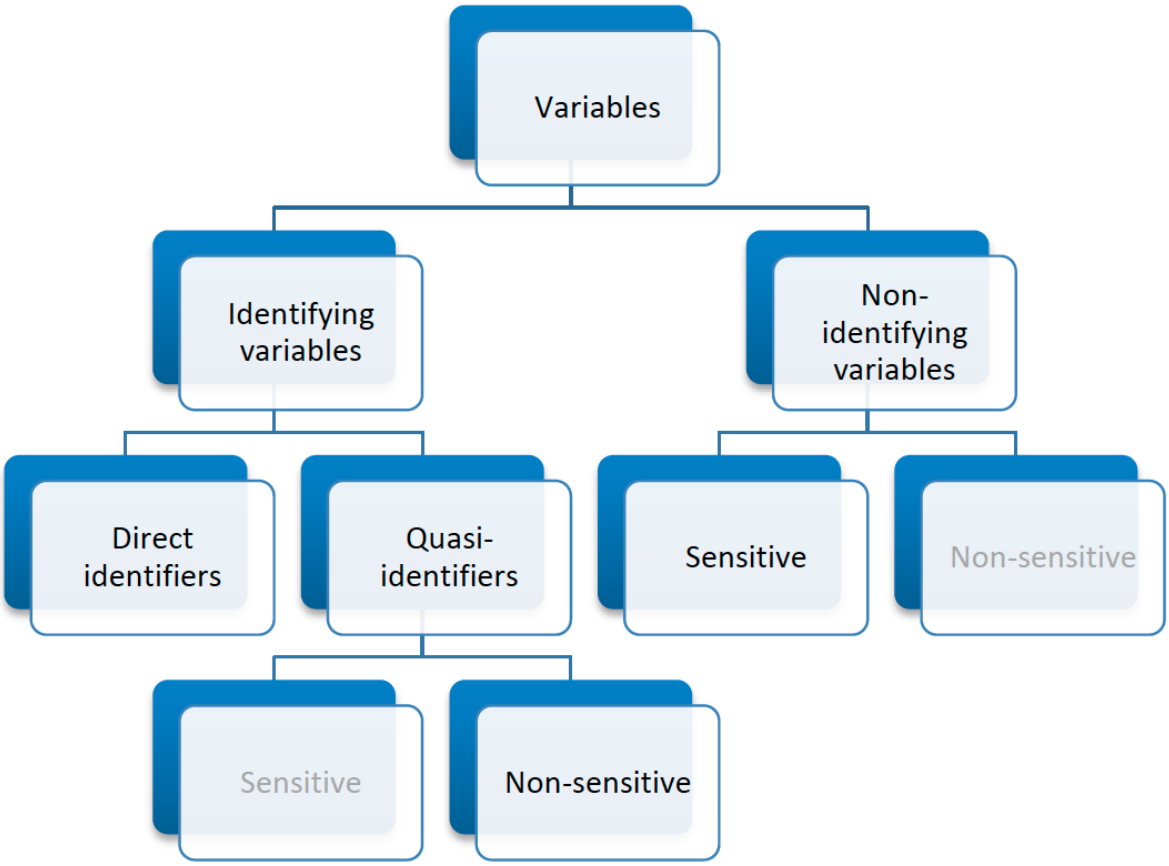

Fig. 5.1: Classification of variables

5.3 Disclosure scenarios

Evaluation of disclosure risk is carried out with reference to the available data sources in the environment where the dataset is to be released. In this setting, disclosure risk is the possibility of correctly re-identifying an individual in the released microdata file by matching their data to an external file based on a set of quasi-identifiers. The risk assessment is done by identifying so-called disclosure or intrusion scenarios. A disclosure scenario describes the information potentially available to the intruder (e.g., census data, electoral rolls, population registers or data collected by private firms) to identify respondents and the ways such information can be combined with the microdata set to be released and used for re-identification of records in the dataset. Typically, these external datasets include direct identifiers. In that case, the re-identification of records in the released dataset leads to identity and, possibly, attribute disclosure. The main outcome of the evaluation of disclosure scenarios is the identification of a set of quasi-identifiers (i.e., key variables) that need to be treated during the SDC process (see *ELMP10*).

An example of a disclosure scenario could be the spontaneous recognition of a respondent by a researcher. For

instance, while going through the data, the researcher recognizes a person with an unusual combination of the variables age and marital status. Of course, this can only happen if the person is well-known or is known to the researcher. Another example of a disclosure scenario for a publicly available file would be if variables in the data could be linked to a publically available electoral register. An intruder might try matching the entire dataset with individuals in the register. However, this might be difficult and take specialized expertise, or software, and other conditions have to be fulfilled. Examples are that the point in time the datasets were collected should approximately match and the content of the variables should be (nearly) identical. If these conditions are not fulfilled, exact matching is much less likely.

Note: Not all external data is necessarily in the public domain. Also privately owned datasets or datasets which are not released should be taken into consideration for determining the suitable disclosure scenario.

Info-box - Disclosure scenarios and different release types

A dataset can have more than one disclosure scenario. Disclosure scenarios also differ depending on the data access type that the data will be released under; for example, Public Use Files (PUF) or Scientific Use Files (SUF, also known as licensed) or in a data enclave. The required level of protection, the potential avenues of disclosure as well as the availability of other external data sources differ according to the access type under which the data will be released. For example, the user of a Scientific Use File (SUF) might be contractually restricted by an agreement as to what they are allowed to do with the data, whereas a Public Use File (PUF) might be freely available on the internet under a much looser set of conditions. PUFs will in general require more protection than SUFs and SUFs will require more protection than those files only released in an data enclave. Disclosure scenarios should be developed with all of this in mind.

The evaluation of disclosure risk is based on the quasi-identifiers, which are identified in the analysis of disclosure risk scenarios. The disclosure risk directly depends on the inclusion or exclusion of variables in the set of quasi-identifiers chosen. This step in the SDC process (making the choice of quasi-identifiers) should therefore be approached with great thought and care. We will see later, as we discuss the steps in the SDC process in more detail, that the first step for any agency is to undertake an exercise in which an inventory is compiled of all datasets available in the country. Both datasets released by the national statistical office and from other sources are considered and their availability to intruders as well as the variables included in these datasets is analyzed. It is this information that will serve as a key metric when deciding which variables to choose as potential identifiers, as well as dictate the level of SDC and methods needed.

5.4 Levels of risk

With microdata from surveys and censuses, we often have to be concerned about disclosure at the individual or unit level, i.e., identifying individual respondents. Individual respondents are generally natural persons, but can also be units, such as companies, schools, health facilities, etc. Microdata files often have a hierarchical structure where individual units belong to groups, e.g., people belong to households. The most common hierarchical structure in microdata is the household structure in household survey data. Therefore, in this guide, we sometimes call disclosure risk for data with a hierarchical structure "household risk". The concepts, however, apply equally to establishment data and other data with hierarchical structures, such as school data with pupils and teachers or company data with employees.

We will see that this hierarchical structure is important to take into consideration when measuring disclosure risk. For hierarchical data, information collected at the higher hierarchical level (e.g., household level) would be the same for all individuals in the group belonging to that higher hierarchical level (e.g., household).[3] Some typical examples of variables that would have the same values for all members of the same higher hierarchical unit are, in the case of

[3] Besides variables collected at the higher hierarchical level, also variables collected at the lower level but with no (or little) variation within the groups formed by the hierarchical structure should be treated as higher level variables. An example could be mother tongue, where most households are monolingual, but the variable is collected at the individual level.

households, those relating to housing and household income. These variables differ from survey to survey and from country to country.[4] This hierarchical structure creates a further level of disclosure risk for two reasons:

1. if one individual in the household is re-identified, the household structure allows for re-identification of the other household members in the same household,
2. values of variables for other household members that are common for all household members can be used for re-identification of another individual of the same household. This is discussed in more detail in the Section *Household Risk*.

Next, we first discuss risk measures used to evaluate disclosure risk in the absence of a hierarchical structure. This includes risk measures that seek to aggregate the individual risk for all individuals in the microdata file; the objective is to quantify a global disclosure risk measure for the file. We then discuss how risk measures change when taking the hierarchical structure of the data into account.

We will also discuss how risk measures differ for categorical and continuous key variables. For categorical variables, we will use the concept of uniqueness of combinations of values of quasi-identifiers (so-called "keys") used to identify individuals at risk. The concept of uniqueness, however, is not useful for continuous variables, since it is likely that all or many individuals will have unique values for that variable, by definition of a continuous variable. Risk measures for categorical variables are generally a priori measures, i.e., they can be evaluated before applying anonymization methods since they are based on the principle of uniqueness. Risk measures for continuous variables are a posteriori measures; they are based on comparing the microdata before and after anonymization and are, for example, based on the proximity of observations between the original and the treated (anonymized) datasets.

Files that are limited to only categorical or only continuous key variables are easiest for risk measurement. We will see in later sections that, in cases where both types of variables are present, recoding of continuous variables into categories is one approach to use to simplify the SDC process, but we will also see that from a utility perspective this may not be desirable. An example might be the use of income quintiles instead of the actual income variables. We will see that measuring the risk of disclosure based on the categorical and continuous variables separately is generally not a valid approach.

The risk measures discussed in the next section are based on several assumptions. In general, these measures rely on quite restrictive assumptions and will often lead to conservative risk estimates. These conservative risk measures may overstate the risk as they assume a worst-case scenario. Two assumptions should, however, be fulfilled for the risk measures to be valid and meaningful; the microdata should be a sample of a larger population (no census) and the sampling weights should be available. The Section Special case: census data briefly discusses how to deal with census data.

5.5 Individual risk

5.5.1 Categorical key variables and frequency counts

The main focus of risk measurement for categorical quasi-identifiers is on identity disclosure. Measuring disclosure risk is based on the evaluation of the probability of correct re-identification of individuals in the released data. We use measures based on the actual microdata to be released. In general, the rarer a combination of values of the quasi-identifiers (i.e., key) of an observation in the sample, the higher the risk of identity disclosure. An intruder that tries to match an individual who has a relatively rare key within the sample data with an external dataset in which the same key exists will have a higher probability of finding a correct match than when a larger number of individuals share the same key. This can be illustrated with the following example that is illustrated in Table 5.1.

Table 5.1 shows values for 10 respondents for the quasi-identifiers "residence", "gender", "education level" and "labor status". In the data, we find seven unique combinations of values of quasi-identifiers (i.e., patterns or keys) of the four

[4] Religion, for example, can be shared by all household members in some countries, whereas in other countries this variable is measured at the individual level and mixed-religion households exist.

quasi-identifiers. Examples of keys are {'urban', 'female', 'secondary incomplete', 'employed'} and {'urban', 'female', 'primary incomplete', 'non-LF'}. Let f_k be the sample frequency of the k^{th} key, i.e., the number of individuals in the sample with values of the quasi-identifiers that coincide with the k^{th} key. This would be 2 for the key {urban, female, secondary incomplete, employed}, since this key is shared by individuals 1 and 2 and 1 for the key {'urban', 'female', 'primary incomplete', 'non-LF'}, which is unique to individual 3. By definition, f_k is the same for each record sharing a particular key.

The fewer the individuals with whom an individual shares his or her combination of quasi-identifiers, the more likely the individual is to be correctly matched in another dataset that contains these quasi-identifiers. Even when direct identifiers are removed from the dataset, that individual has a higher disclosure risk than others, assuming that their sample weights are the same. Table 5.1 reports the sample frequencies f_k of the keys for all individuals. Individuals with the same keys have the same sample frequency. If $f_k = 1$, this individual has a unique combination of values of quasi-identifiers and is called "sample unique". The dataset in Table 5.1 contains four sample uniques. Risk measures are based on this sample frequency.

Table 5.1: Example dataset showing sample frequencies, population frequencies and individual disclosure risk

No	Residence	Gender	Education level	Labor status	Weight	f_k	F_k	risk
1	Urban	Female	Secondary incomplete	Employed	180	2	360	0.0054
2	Urban	Female	Secondary incomplete	Employed	180	2	360	0.0054
3	Urban	Female	Primary incomplete	Non-LF	215	1	215	0.0251
4	Urban	Male	Secondary complete	Employed	76	2	152	0.0126
5	Rural	Female	Secondary complete	Unemployed	186	1	186	0.0282
6	Urban	Male	Secondary complete	Employed	76	2	152	0.0126
7	Urban	Female	Primary complete	Non-LF	180	1	180	0.0290
8	Urban	Male	Post-secondary	Unemployed	215	1	215	0.0251
9	Urban	Female	Secondary incomplete	Non-LF	186	2	262	0.0074
10	Urban	Female	Secondary incomplete	Non-LF	76	2	262	0.0074

In Listing 5.1, we show how to use the *sdcMicro* package to create a list of sample frequencies f_k for each record in a dataset. This is done by using the *sdcMicro* function freq(). A value of 2 for an observation means that in the sample, there is one more individual with exactly the same combination of values for the selected key variables. In Listing 5.1, the function freq() is applied to "sdcInitial", which is an *sdcMicro* object. Footnote[5] shows how to initialize the *sdcMicro* object for this example. For a complete discussion of *sdcMicro* objects as well as instructions on how to create *sdcMicro* objects, we refer to the section Objects of class sdcMicroObj. *sdcMicro* objects are used when doing SDC with *sdcMicro*. The function freq() displays the sample frequency for the keys constructed on a defined set of quasi-identifiers. Listing 5.1 corresponds to the data in Table 5.1.

[5] The code examples in this guide are based on *sdcMicro* objects. An *sdcMicro* object contains, amongst others, the data and identifies all the specified key variables. The code below creates a data.frame with the data from Table 5.1 and the *sdcMicro* objects "sdcInitial" used in most examples in this section.

Listing 5.1: Calculating f_k using *sdcMicro*

```
# Frequency of the particular combination of key variables (keys) for each record in the sample
freq(sdcInitial, type = 'fk')
2 2 1 2 1 2 1 1 2 2
```

For sample data, it is more interesting to look at F_k, the population frequency of a combination of quasi-identifiers (key) k, which is the number of individuals in the population with the key that corresponds to key k. The population frequency is unknown if the microdata is a sample and not a census. Under certain assumptions, the expected value of the population frequencies can be computed using the sample design weight w_i (in a simple sample, this is the inverse of the inclusion probability) for each individual i

$$F_k = \sum_{i|key\ of\ individual\ i\ corresponds\ to\ key\ k} w_i$$

F_k is the sum of the sample weights of all records with the same key k. Hence, like f_k, F_k is the same for each record with key k. The risk of correct re-identification is the probability that the key is matched to the correct individual in the population. Since every individual in the sample with key k corresponds to F_k individuals in the population, the probability of correct re-identification is $1/F_k$. This is the probability of re-identification in the worst-case scenario and can be interpreted as disclosure risk. Individuals with the same key have the same frequencies, i.e., the frequency of the key.

If $F_k = 1$, the key k is both a sample and a population unique and the disclosure risk would be 1. Population uniques are an important factor to consider when evaluating risk, and deserve special attention. Table 5.1 also shows F_k for the example dataset. This is further discussed in the case studies the Section Case Studies.

Besides f_k, the sample frequency of key k (i.e., the number of individuals in the sample with the combination of quasi-identifiers corresponding to the combination specified in key k) and F_k, the estimated population frequency of key k, can be displayed in *sdcMicro*. Listing 5.2 illustrates how to return lists of length n of frequencies for all individuals. The frequencies are displayed for each individual and not for each key.

```
library(sdcMicro)

# Set up dataset
data <- as.data.frame(cbind(as.factor(c('Urban', 'Urban', 'Urban', 'Urban', 'Rural', 'Urban', 'Urban', 'Urban',
                                        'Urban', 'Urban')),
                            as.factor(c('Female', 'Female', 'Female', 'Male',
                                        'Female', 'Male', 'Female', 'Male', 'Female', 'Female')),
                            as.factor(c('Sec in', 'Sec in', 'Prim in', 'Sec com', 'Sec com', 'Sec com', 'Prim com', 'Post-sec', 'Sec in', 'Sec in')),
                            as.factor(c('Emp', 'Emp', 'Non-LF', 'Emp', 'Unemp', 'Emp', 'Non-LF', 'Unemp', 'Non-LF','Non-LF')),
                            as.factor(c('yes', 'yes', 'yes', 'yes', 'yes', 'no', 'no', 'yes', 'no', 'yes')),
                           c(180, 180, 215, 76, 186, 76, 180, 215, 186, 76)
                           ))

# Specify variable names
names(data) <- c('Residence', 'Gender', 'Educ', 'Lstat', 'Health', 'Weights')

# Set up sdcMicro object with specified quasi-identifiers and weight variable
sdcInitial <- createSdcObj(dat = data, keyVars = c('Residence', 'Gender', 'Educ', 'Lstat'), weightVar = 'Weights')
```

Listing 5.2: Calculating the sample and population frequencies using *sdcMicro*

```
# Sample frequency of individual's key
freq(sdcInitial, type = 'fk')
2 2 1 2 1 2 1 1 2 2

# Population frequency of individual's key
freq(sdcInitial, type = 'Fk')
360 360 215 152 186 152 180 215 262 262
```

In practice, this approach leads to conservative risk estimates, as it does not adequately take the sampling methods into account. In this case, the estimates of re-identification risk may be estimated too high. If this overestimated risk is used, the data may be overprotected (i.e., information loss will be higher than was necessary) when applying SDC measures. Instead, a Bayesian approach to risk measurement is recommended, where the posterior distribution of F_k is used (see e.g., *HDFG12*) to estimate an individual risk measure r_k for each key k.

The risk measure r_k is, as f_k and F_k, the same for all individuals sharing the same pattern of values of key variables and is referred to as individual risk. The values r_k can also be interpreted as the probability of disclosure for the individuals or as the probability for a successful match with individuals chosen at random from an external data file with the same values of the key variables. This risk measure is based on certain assumptions[6], which are strict and may lead to a relatively conservative risk measure. In *sdcMicro*, the risk measure r_k is automatically computed when creating an *sdcMicro* object and saved in the "risk" slot[7]. Listing 5.3 shows how to retrieve the risk measures using *sdcMicro* for our example. The risk measures are also presented in Table 5.1.

Listing 5.3: The individual risk slot in the *sdcMicro* object

```
    sdcInitial@risk$individual

              risk           fk    Fk
    [1,]    0.005424520    2     360
    [2,]    0.005424520    2     360
    [3,]    0.025096439    1     215
    [4,]    0.012563425    2     152
    [5,]    0.028247279    1     186
    [6,]    0.012563425    2     152
    [7,]    0.029010932    1     180
    [8,]    0.025096439    1     215
    [9,]    0.007403834    2     262
    [10,]   0.007403834    2     262
```

The main factors influencing the individual risk are the sample frequencies f_k and the sampling design weights w_i. If an individual is at relatively high risk of disclosure, in our example this would be individuals 3, 5, 7 and 8 in Table 5.1 and Listing 5.3, the probability that a potential intruder correctly matches these individuals with an external data file is high **relative to the other individuals in the released data.** In our example, the reason for the high risk is the fact that these individuals are sample uniques ($f_k = 1$). This risk is the worst-case scenario risk and does not imply that the person will be re-identified with certainty with this probability. For instance, if an individual included in the microdata is not included in the external data file, the probability for a correct match is zero. Nevertheless, the risk measure computed based on the frequencies will be positive.

[6] The assumptions for this risk measure are strict and the risk is estimated in many cases higher than the actual risk. Among other assumptions, it is assumed that all individuals in the sample are also included in the external file used by the intruder to match against. If this is not the case, the risk is much lower; if the individual in the released file is not contained in the external file, the probability of a correct match is zero. Other assumptions are that the files contain no errors and that both sets of data were collected simultaneously, i.e. they contain the same information. These assumptions will often not hold generally, but are necessary for computation of a measure. An example of a violation of the last assumptions is could occur if datasets are collected at different points in time and records have changed. This could happen when people move or change jobs and makes correct matching impossible. The assumptions assume a worst-case scenario.

[7] See the Section Objects of class sdcMicroObj for more information on slots and the *sdcMicro* object structure.

5.5.2 k-anonymity

The risk measure k-anonymity is based on the principle that, in a safe dataset, the number of individuals sharing the same combination of values (keys) of categorical quasi-identifiers should be higher than a specified threshold k. k-anonymity is a risk measure based on the microdata to be released, since it only takes the sample into account. An individual violates k-anonymity if the sample frequency count f_k for the key k is smaller than the specified threshold k. For example, if an individual has the same combination of quasi-identifiers as two other individuals in the sample, these individuals satisfy 3-anonymity but violate 4-anonymity. In the dataset in Table 5.1, six individuals satisfy 2-anonymity and four violate 2-anonymity. The individuals that violate 2-anonymity are sample uniques. The risk measure is the number of observations that violates k-anonymity for a certain value of *k*, which is

$$\sum_i I(f_k < k),$$

where I is the indicator function and i refers to the i^{th} record. This is simply a count of the number of individuals with a sample frequency of their key lower than k. The count is higher for larger k, since if a record satisfies k-anonimity, it also satisfies $(k+1)$-anonimity. The risk measure k-anonymity does not consider the sample weights, but it is important to consider the sample weights when determining the required level of k-anonymity. If the sample weights are large, one individual in the dataset represents more individuals in the target population, the probability of a correct match is smaller, and hence the required threshold can be lower. Large sample weights go together with smaller datasets. In a smaller dataset, the probability to find another record with the same key is smaller than in a larger dataset. This probability is related to the number of records in the population with a particular key through the sample weights.

In *sdcMicro* we can display the number of observations violating a given k-anonymity threshold. In Listing 5.4, we use *sdcMicro* to calculate the number of violators for the thresholds $k = 2$ and $k = 3$. Both the absolute number of violators and the relative number as percentage of the number of individuals in the sample are given. In the example, four observations violate 2-anonimity and all 10 observations violate 3-anonymity.

Listing 5.4: Using the print() function to display observations violating k-anonymity

```
print(sdcInitial, 'kAnon')

Number of observations violating
-  2-anonymity: 4
-  3-anonymity: 10
---------------------------
Percentage of observations violating
-  2-anonymity: 40 %
-  3-anonymity: 100 %
```

For other levels of k-anonymity, it is possible to compute the number of violating individuals by using the sample frequency counts in the *sdcMicro* object. The number of violators is the number of individuals with sample frequency counts smaller than the specified threshold k. In Listing 5.5, we show an example of how to calculate any threshold for k using the already-stored risk measures available after setting up an *sdcMicro* object in *R*. k can be replaced with any required threshold. The choice of the required threshold that all individuals in the microdata file should satisfy depends on many factors and is discussed further in the Section Local suppression on local suppression. In many institutions, typically required thresholds for k-anonymity are 3 and 5.

Listing 5.5: Computing k-anonymity violations for other values of k

```
sum(sdcInitial@risk$individual[,2] < k)
```

It is important to note that missing values ('NA's in *R*[8]) are treated as if they were any other value. Two individuals with keys {'Male', NA, 'Employed'} and {'Male', 'Secondary complete', 'Employed'} share the same key, and similarly, {'Male', NA, 'Employed'} and {'Male', 'Secondary incomplete', 'Employed'} also share the same key. Therefore, the missing value in the first key is first interpreted as 'Secondary complete', and then as 'Secondary incomplete'. This is illustrated in Table 5.2.

Note: The sample frequency of the third record is 3, since it is regarded to share its key both with the first and second record.

This principle is used when applying local suppression to achieve a certain level of k-anonymity (see the Section Local suppression) and is based on the fact that the value NA could replace any value.

Table 5.2: Example dataset to illustrate the effect of missing values on k-anonymity

No	Gender	Education level	Labor status	f_k
1	Male	Secondary complete	Employed	2
2	Male	Secondary incomplete	Employed	2
3	Male	NA	Employed	3

If a dataset satisfies k-anonymity, an intruder will always find at least k individuals with the same combination of quasi-identifiers. k-anonymity is often a necessary requirement for anonymization for a dataset before release, but is not necessarily a sufficient requirement. The k-anonymity measure is only based on frequency counts and does not take (differences in) sample weights into account. Often k-anonymity is achieved by first applying recoding and subsequently applying local suppression, and in some cases by microaggregation, before using other risk measures and disclosure methods to further reduce disclosure risk. These methods are discussed in the Section Anonymization methods.

5.5.3 l-diversity

k-anonymity has been criticized for not being restrictive enough. Sensitive information might be disclosed even if the data satisfies k-anonymity. This might occur in cases where the data contains sensitive (non-identifying) categorical variables that have the same value for all individuals that share the same key. Examples of such sensitive variables are those containing information on an individual's health status. Table 5.3 illustrates this problem by using the same data as previously used, but adding a sensitive variable, "health". The first two individuals satisfy 2-anonymity for the key variables "residence", "gender", "education level" and "labor status". This means that an intruder will find at least two individuals when matching to the released microdata set based on those four quasi-identifiers. Nevertheless, if the intruder knows that someone belongs to the sample and has the key {'Urban', 'Female', 'Secondary incomplete' and 'Employed'}, with certainty the health status is disclosed ('yes'), because both observations with this key have the same value. This information is thus disclosed without the necessity to match exactly to the individual. This is not the case for the individuals with the key {'Urban', 'Male', 'Secondary complete', 'Employed'}. Individuals 4 and 6 have different values ('yes' and 'no') for "health", and thus the intruder would not gain information about the health status from this dataset by matching an individual to one of these individuals.

[8] In *sdcMicro* it is important to use the standard missing value code NA instead of other codes, such as 9999 or strings. In the Section Missing values, we further discuss how to set other missing value codes to NA in *R*. This is necessary to ensure that the methods in *sdcMicro* function properly. When missing values have codes other than NA, the missing value codes are interpreted as a distinct factor level in the case of categorical variables.

Table 5.3: l-diversity illustration

No	Residence	Gender	Education level	Labor status	Health	f_k	F_k	l-diversity
1	Urban	Female	Secondary incomplete	Employed	yes	2	360	1
2	Urban	Female	Secondary incomplete	Employed	yes	2	360	1
3	Urban	Female	Primary incomplete	Non-LF	yes	1	215	1
4	Urban	Male	Secondary complete	Employed	yes	2	152	2
5	Rural	Female	Secondary complete	Unemployed	yes	1	186	1
6	Urban	Male	Secondary complete	Employed	no	2	152	2
7	Urban	Female	Primary complete	Non-LF	no	1	180	1
8	Urban	Male	Post-secondary	Unemployed	yes	1	215	1
9	Urban	Female	Secondary incomplete	Non-LF	no	2	262	2
10	Urban	Female	Secondary incomplete	Non-LF	yes	2	262	2

The concept of (distinct) l-diversity addresses this shortcoming of k-anonymity (see *MKGV07*). A dataset satisfies l-diversity if for every key k there are at least l different values for each of the sensitive variables. In the example, the first two individuals satisfy only 1-diversity, individuals 4 and 6 satisfy 2-diversity. The required level of l-diversity depends on the number of possible values the sensitive variable can take. If the sensitive variable is a binary variable, the highest level if l-diversity that can be achieved is 2. A sample unique will always only satisfy 1-diversity.

To compute l-diversity for sensitive variables in *sdcMicro*, the function ldiversity() can be used. This is illustrated in Listing 5.6. As arguments, we specify the names of the sensitive variables[9] in the file as well as a constant for recursive l-diversity,[10] and the code for missing values in the data. The output is saved in the "risk" slot of the *sdcMicro* object. The result shows the minimum, maximum, mean and quantiles of the l-diversity scores for all individuals in the sample. The output in Listing 5.6 reproduces the results based on the data in Table 5.3.

Listing 5.6: l-diversity function in *sdcMicro*

```
# Computing l-diversity

sdcInitial <- ldiversity(obj = sdcInitial, ldiv_index = c("Health"), l_recurs_c = 2, missing = NA)
# Output for l-diversity
sdcInitial@risk$ldiversity

--------------------------
L-Diversity Measures
--------------------------
Min.  1st Qu.  Median    Mean   3rd Qu.    Max.
1.0   1.0      1.0       1.4    2.0        2.0

# l-diversity score for each record
sdcInitial@risk$ldiversity[,'Health_Distinct_Ldiversity']

[1] 1 1 1 2 1 2 1 1 2 2
```

l-diversity is useful if the data contains categorical sensitive variables that are not quasi-identifiers themselves. It is not possible to select quasi-identifiers to calculate the l-diversity. l-diversity has to be calculated for each sensitive variable separately.

[9] Alternatively, the sensitive variables can be specified when creating the *sdcMicro* object using the function createSdcObj() in the *sensibleVar* argument. This is further explained in the Section Objects of class sdcMicroObj . In that case, the argument *ldiv_index* does not have to be specified in the ldiversity() function. and the variables in the *sensibleVar* argument will automatically be used to compute l-diversity.

[10] Besides distinct l-diversity, there are other l-diversity methods: entropy and recursive. Distinct l-diversity is most commonly used.

5.6 Special Uniques Detection Algorithm (SUDA)

The previously discussed risk measures depend on identifying key variables for which there may be information available from other sources or other datasets, and which, when combined with the current data, may lead to re-identification. In practice, however, it might not always be possible to conduct an inventory of all available datasets and their variables and thus assess all known external linkages and risks.

To overcome this, an alternative heuristic measure based on special uniques has been developed to determine the riskiness of a record, which leads to a SUDA metric or score (see *ElMF02*). These measures are based on the search for special uniques. To find these special uniques, algorithms, called SUDA (Special Uniqueness Detection Algorithm), have been developed. SUDA algorithms are based on the concept of special uniqueness, which is introduced in the next subsection. Since this is a heuristic approach, its performance is only tested in actual datasets, which is done in *ElMF02* for UK census data. These tests have shown that the performance of the algorithm leads to good risk estimates for these test datasets.

5.6.1 Sample unique vs. special unique

The previous measures of risk focused on the uniqueness of the key of a record in the dataset. Table 5.4 reproduces the data from Table 5.1. The sample dataset has 10 records and four pre-determined quasi-identifiers {"Residence", "Gender", "Education level" and "Labor status"}. Given the four quasi-identifiers, we have seven distinct patterns in those key variables, or keys (e.g., {'Urban', 'Female', 'Secondary incomplete', 'Employed'}). The sample frequency counts of the first and second records equal 2, because the two records share the same pattern (i.e., {'Urban', 'Female', 'Secondary incomplete', 'Employed'}). Record 3 is a sample unique because it is the only individual in the sample who is a female living in an urban area who is employed without completing primary school. Similarly, records 5, 7 and 8 are sample uniques, because they possess distinct patterns with respect to the four key variables.

Table 5.4: Sample uniques and special uniques

No	Residence	Gender	Education level	Labor status	Weight	f_k	F_k	risk
1	Urban	Female	Secondary incomplete	Employed	180	2	360	0.0054
2	Urban	Female	Secondary incomplete	Employed	180	2	360	0.0054
3	Urban	Female	Primary incomplete	Non-LF	215	1	215	0.0251
4	Urban	Male	Secondary complete	Employed	76	2	152	0.0126
5	Rural	Female	Secondary complete	Unemployed	186	1	186	0.0282
6	Urban	Male	Secondary complete	Employed	76	2	152	0.0126
7	Urban	Female	Primary complete	Non-LF	180	1	180	0.0290
8	Urban	Male	Post-secondary	Unemployed	215	1	215	0.0251
9	Urban	Female	Secondary incomplete	Non-LF	186	2	262	0.0074
10	Urban	Female	Secondary incomplete	Non-LF	76	2	262	0.0074

In addition to the records 3, 5, 7 and 8 in Table 5.4 being sample uniques with respect to the key variable set {"Residence", "Gender", "Education level", "Labor status"}, we can find unique patterns in these records without even having to consider the complete set of key variables. For instance, a unique pattern can be found in record 5 when we look only at the variables "Education level" and "Labor status" ({'Secondary complete', 'Unemployed'}). While the values {'Secondary complete'} and {'Unemployed'} are not unique in the sample, the combination of them, {'Secondary complete', 'Unemployed'} makes record 5 unique. This variable subset is referred to as the Minimal Sample Unique (MSU) as any smaller subset of this set of variables is not unique (in this case {'Secondary complete'} and {'Unemployed'}). It is an MSU of size 2. This holds as well for three other combinations in record 5, i.e., {'Female', 'Unemployed'} and {'Female', 'Secondary Complete'}, which are also MSUs of size 2 and {'Rural'} of size 1. In total, record 5 has four MSUs[11]. To determine if a set is an MSU of size k, we check whether it fulfills the minimal

[11] There are more combinations of quasi-identifiers that make record 5 unique (e.g., {'Rural', 'Female'} and {'Female', 'Secondary Complete', 'Unemployed'}. These combinations, however, are not considered MSUs because they do not fulfill the **minimal** subset requirement. They contain subsets that are MSUs.

requirement. It suffices to check whether all subsets of size $k-1$ of the MSU are unique. If any of these subsets are also unique in the sample, the set found may be a sample unique, but violates the minimal requirement and is hence not an MSU. The unique subset of size $k-1$ could be a MSU. In our example, to determine if the MSU {'Secondary complete', 'Unemployed'} is a MSU, we checked as to whether its subsets {'Secondary complete'} and {'Unemployed'} were not unique in the sample. By definition, only sample uniques can be special uniques.

The SUDA algorithm identifies all the MSUs in the sample, which in turn are used to assign a SUDA score to each record. This score indicates how "risky" a record is. The potential risk of the records is determined based on two observations:

- The smaller the size of the MSU within a record (i.e., the fewer variables are needed to reach uniqueness), the greater the risk of the record
- The larger the number of MSUs possessed by a record, the greater the risk of the record

A record is defined as a special unique if it is a sample unique both on the complete set of quasi-identifiers (e.g., in the data in Table 5.4, the variables "Residence", "Gender", "Education level" and "Labor status") and simultaneously has at least one MSU (*ElSD98*). Special uniques can be classified according to the number and size of subsets that are MSUs. Research has shown that special uniques are more likely to be population uniques than random uniques (*ElMF02*) and are thus relevant for risk assessment.

5.6.2 Calculating SUDA scores

The SUDA algorithm is used to search for MSUs in the data among the sample uniques to determine which sample uniques are also special uniques i.e., have subsets that are also unique (see Elliot et al., 2005). First the SUDA algorithm is used to identify the MSUs for each sample unique. To simplify the search and because smaller subsets are more important for disclosure risk, the search is limited to a maximum subset size. Subsequently, a score is assigned to each individual, which ranks the individuals according to their level of risk.

For each MSU of size k contained in a given record, a score is computed by $\prod_{i=k}^{M}(ATT-i)$, where M is the user-specified maximum size of MSUs[12], and ATT is the total number of attributes or variables in the dataset. By definition, the smaller the size k of the MSU, the larger the score for the MSU, which reflects greater risk (see *EMMG05*). The final SUDA score for each record is computed by adding the scores for each MSU in the record. In this way, records with more MSUs are assigned a higher SUDA score, which also reflects the higher risk. The SUDA score ranks the individuals according to their level of risk. The higher the SUDA score, the riskier the sample unique.

Calculating SUDA scores – a simplified example

To illustrate how SUDA scores are calculated, we compute the SUDA scores for the sample uniques in the data in Table 5.5, which replicates the data from Table 5.5. Record 5 contains four MSUs: {Rural} of size 1, and {'Secondary Complete', 'Unemployed'}, {'Female', 'Unemployed'} and {Female, Secondary Complete} of size 2. Suppose the maximum size of MSUs we search for in the data, M, is set at 3. Knowing that, ATT, the number of selected key variables in the dataset, is 4; the score assigned to {Rural} is computed by $\prod_{i=1}^{3}(4-i) = 3*2*1 = 6$; and the score assigned to {Secondary complete, Unemployed}, {Female, Unemployed} and {Female, Secondary Complete} is $\prod_{i=2}^{3}(4-i) = 2*1 = 2$. The SUDA score for the fifth record in Table 5.5 is then $6+2+2+2 = 12$, which is the sum of these four scores per MSU. The SUDA scores for the other sample uniques are computed accordingly[13]. The values that are in the MSUs in the sample uniques are shaded in Table 5.5. Records that are not sample uniques ($f_k > 1$) cannot be special uniques and are assigned the score 0.

[12] OECD, http://stats.oecd.org/glossary

[13] The third record has one MSU, {'Primary incomplete'}; the seventh record has one MSU, {'Primary complete'}; and the eighth record has three MSUs, {'Urban, Unemployed'}, {'Male, Unemployed'} and {'Post-secondary'}.

Table 5.5: Illustrating the calculation of SUDA and DIS-SUDA scores

No	Residence	Gender	Education level	Labor status	Weight	f_k	SUDA score	DIS-SUDA
1	Urban	Female	Secondary incomplete	Employed	180	2	0	0.0000
2	Urban	Female	Secondary incomplete	Employed	180	2	0	0.0000
3	Urban	Female	Primary incomplete	Non-LF	215	1	6	0.0051
4	Urban	Male	Secondary complete	Employed	76	2	0	0.0000
5	Rural	Female	Secondary complete	Unemployed	186	1	12	0.0107
6	Urban	Male	Secondary complete	Employed	76	2	0	0.0000
7	Urban	Female	Primary complete	Non-LF	180	1	6	0.0051
8	Urban	Male	Post-secondary	Unemployed	215	1	10	0.0088
9	Urban	Female	Secondary incomplete	Non-LF	186	2	0	0.0000
10	Urban	Female	Secondary incomplete	Non-LF	76	2	0	0.0000

To estimate record-level disclosure risks, SUDA scores can be used in combination with the Data Intrusion Simulation (DIS) metric (*ElMa03*) , a method for assessing disclosure risks for the entire dataset (i.e., file-level disclosure risks). Roughly speaking, the DIS-SUDA method distributes the file-level risk measure generated by the DIS metric between records according to the SUDA scores of each record. This way, SUDA scores are calibrated against a consistent measure to produce the DIS-SUDA scores, which provide the record-level disclosure risk. These scores are used to compute the conditional probability that a unique match found by an intruder between the sample unique in the released microdata and an external data source is also a correct match, and hence a successful disclosure. The DIS-SUDA measure can be computed in *sdcMicro*. Since the DIS score is a probability, its values are in the interval $[0,\ 1]$. A full description of the DIS-SUDA method is provided by *ElMa03*.

Note that unlike the risk methods discussed earlier, the DIS-SUDA score does not fully account for the sampling weights. Risk measures based on the previous methods (i.e., negative binomial models) will in general have lower risks for those records with greater sampling weight, given the same sample frequency count, than those measured using DIS-SUDA. Therefore, instead of replacing the risk measures introduced in the previous section, the SUDA scores and DIS-SUDA approach should be used as a complementary method. As mentioned earlier, DIS-SUDA is particularly useful in situations where taking an inventory of all already available datasets and their variables is difficult.

5.6.3 Application of SUDA, DIS-SUDA using *sdcMicro*

Both SUDA and DIS-SUDA scores can be computed using *sdcMicro* (*TMKC14*). Given that the search for MSUs with the SUDA algorithm can be computationally demanding, *sdcMicro* uses an improved SUDA2 algorithm, which more effectively locates the boundaries of the search space for MSUs (*MaHK08*).

SUDA scores can be calculated using the suda2() function in *sdcMicro*. It is important to specify the missing argument in suda2(). This should match the code for missing values in your dataset. In *R* this is most likely the *R* standard missing value, NA. We mention this because **the default missing value code in the sdcMicro suda2() function is -999 and will most likely need to be changed to 'NA' when using most R datasets.** The scores are saved in the risk slot of the *sdcMicro* object. The syntax in Listing 5.7 shows how to retrieve the output.

Listing 5.7: Evaluating SUDA scores

```
# Evaluating SUDA scores for the specified variables
sdcInitial <- suda2(obj = sdcInitial, missing = NA)

# The results are saved in the risk slot of the sdcMicro object
# SUDA scores
sdcInitial@risk$suda2$score
[1] 0.00 0.00 1.75 0.00 3.25 0.00 1.75 2.75 0.00 0.00

# DIS-SUDA scores
sdcInitial@risk$suda2$disScore
[1] 0.000000000 0.000000000 0.005120313 0.000000000 0.010702061
[6] 0.000000000 0.005120313 0.008775093 0.000000000 0.000000000

# Summary of DIS-SUDA scores
sdcInitial@risk$suda2

Dis suda scores table:
- - - - - - - - - - -
thresholds number
1         > 0      6
2       > 0.1      4
3       > 0.2      0
4       > 0.3      0
5       > 0.4      0
6       > 0.5      0
7       > 0.6      0
8       > 0.7      0
- - - - - - - - - - -
```

To compare DIS scores before and after applying SDC methods, it may be useful to use histograms or density plots of these scores. Listing 5.8 shows how to generate histograms of the SUDA scores summarized in Listing 5.7. The histogram is shown in Fig. 5.2. All outputs relate to the data used in the example. In our case, we have not applied any SDC method to the data yet and thus have only the plots for the initial values. Typically, after applying SDC methods, one would recalculate the SUDA scores and compare them to the original values. One way to quickly see the differences would be to rerun these visualizations and compare them to the base for risk changes.

Listing 5.8: Histogram and density plots of DIS-SUDA scores

```
# Plot a histogram of disScore
hist(sdcInitial@risk$suda2$disScore, main = 'Histogram of DIS-SUDA scores')

# Density plot
density <- density(sdcInitial@risk$suda2$disScore)
plot(density, main = 'Density plot of DIS-SUDA scores')
```

5.7 Risk measures for continuous variables

The principle of rareness or uniqueness of combinations of quasi-identifiers (keys) is not useful for continuous variables, because it is likely that all or many individuals will have unique keys. Therefore, other approaches are exploited for measuring the disclosure risk of continuous variables. These methods are based on uniqueness of the values in the neighborhood of the original values. The uniqueness is defined in different ways: in absolute terms (interval measure) or relative terms (record linkage). Most measures are a posteriori measures: they are evaluated after anonymization

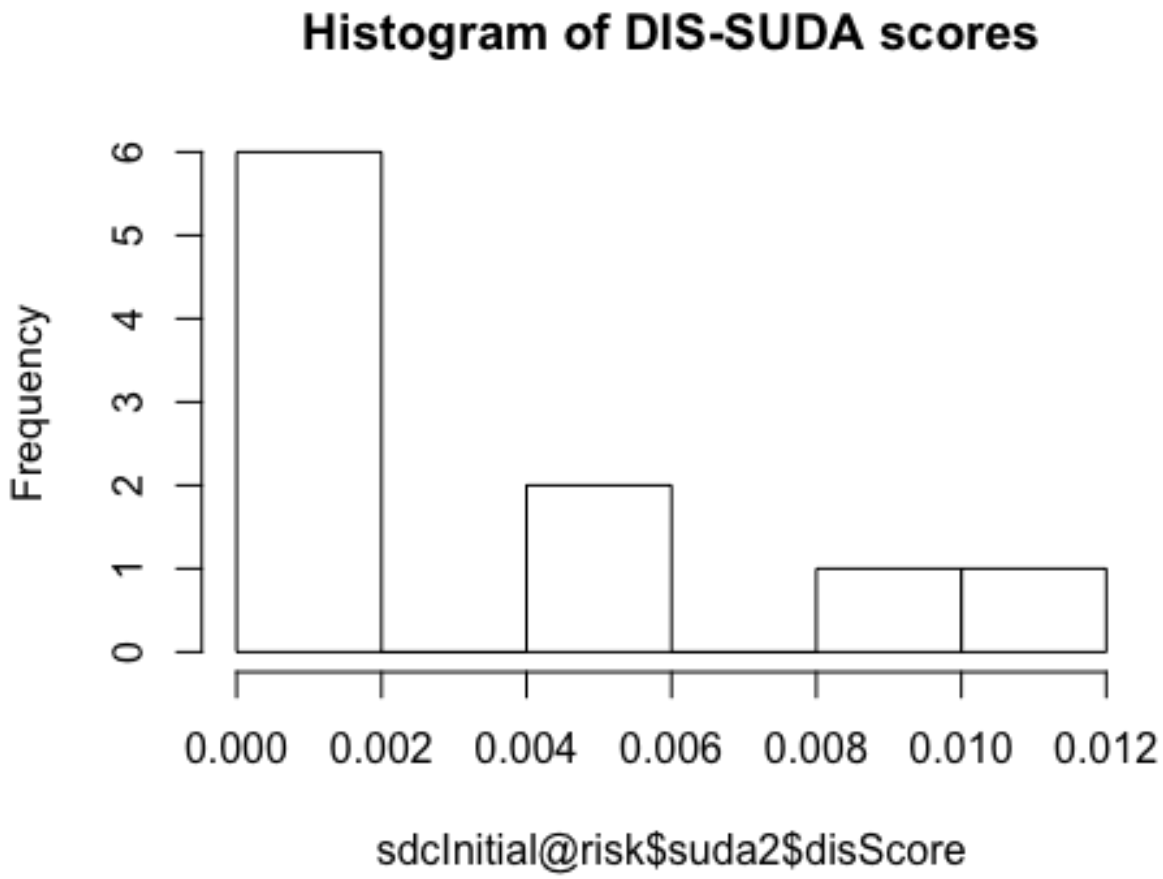

Fig. 5.2: Visualizations of DIS-SUDA scores

of the raw data, compare the treated data with the raw data and evaluate for each individual the distance between the values in the raw and the treated data. This means that these methods are not useful for identifying individuals at risk within the raw data, but rather show the distance/difference between the dataset before and after anonymization and can therefore be interpreted as evaluation of the anonymization method. For that reason, they resemble the information loss measures discussed in the Section Measuring utility and information loss. Finally, risk measures for continuous quasi-identifiers are also based on outlier detection. Outliers play an important role in the re-identification of these records.

5.7.1 Record linkage

Record linkage is an a posteriori method that evaluates the number of correct linkages when linking the perturbed values with the original values. The linking algorithm is based on the distance between the original and the perturbed values (i.e., distance-based record linkage). The perturbed values are matched with the closest individual. It is important to note that this method does not give information on the initial risk, but is rather a measure to evaluate the perturbation algorithm (i.e., it is designed to indicate the level of uncertainty introduced into the variable by counting the number of records that could be correctly matched).

Record linkage algorithms differ with respect to which distance measure is used. When a variable has very different scaling than other continuous variables in the dataset, rescaling the variables before using record linkage is recommended. Very different scales may lead to undesired results when measuring the multivariate distance between records based on several continuous variables. Since these methods are based on both the raw data and treated data, examples of their applications require the introduction of SDC methods and are therefore postponed to the case studies in the Section Case Studies.

Besides distance-based record linkage, another method for linking is probabilistic record linkage (see *DoTo03*). The literature shows, however, that results from distance-based record linkage are better than the results from probabilistic record linkage. Individuals in the treated data that are linked to the correct individuals in the raw data are considered at risk of disclosure.

5.7.2 Interval measure

Successful application of an SDC method should result in perturbed values that are considered not too close to their initial values; if the value is relatively close, re-identification may be relatively easy. In the application of interval measures, intervals are created around each perturbed value and then a determination is made as to whether the original value of that perturbed observation is contained in this interval. Values that are within the interval around the initial value after perturbation are considered too close to the initial value and hence unsafe and need more perturbation. Values that are outside of the intervals are considered safe. The size of the intervals is based on the standard deviation of the observations and a scaling parameter. This method is implemented in the function dRisk() in *sdcMicro*. Listing 5.9 shows how to print or display the risk value computed by *sdcMicro* by comparing the income variables before and after anonymization. "sdcObj" is an *sdcMicro* object and "compExp" is a vector containing the names of the income variables. The size of the intervals is k times the standard deviation, where k is a parameter in the function dRisk(). The larger k, the larger the intervals are, and hence the larger the number of observations within the interval constructed around their original values and the higher the risk measure. The result 1 indicates that all (100 percent) the observations are outside the interval of 0.1 times the standard deviation around the original values.

Listing 5.9: Example with the function dRisk()

```
dRisk(obj = sdcObj@origData[,compExp], xm = sdcObj@manipNumVars[,compExp], k = 0.1)
[1] 1
```

For most values, this is a satisfactory approach. It is not a sufficient measure for outliers, however. After perturbation, outliers will stay outliers and are easily re-identifiable, even if they are sufficiently far from their initial values. Therefore, outliers should be treated with caution.

5.7.3 Outlier detection

Outliers are important for measuring re-identification risk in continuous microdata. Continuous data are often skewed, especially right-skewed. This means that there are a few outliers with very high values relative to the other observations of the same variable. Examples are income in household data, where only few individuals/households may have very high incomes, or turnover data for firms that are much larger than other firms in the sample are. In cases like these, even if these values are perturbed, it may still be easy to identify these outliers, since they will stay the largest values even after perturbation. (The perturbation will have created uncertainty as to the exact value, but because the value started out so much further away from other observations, it may still be easy to link to the high-income individual or very large firm.) Examples would be the only doctor in a geographical area with a high income or one single large firm in one industry type. Therefore, identifying outliers in continuous data is an important step when identifying individuals at high risk. In practice, identifying the values of a continuous variable that are larger than a predetermined $p\%$-percentile might help identify outliers, and thus units at greater risk of identification. The value of p depends on the skewness of the data.

We can calculate the $p\%$-percentile of a continuous variable in *R* and show the individuals who have income larger than this percentile. Listing 5.10 provides an illustration for the 90$^{\text{th}}$ percentile.

Listing 5.10: Computing 90 % percentile of variable income

```
# Compute the 90 % percentile for the variable income
perc90 <- quantile(file[,'income'], 0.90, na.rm = TRUE)

# Show the ID of observations with values for income larger than the 90 % percentile
file[(file[, 'income'] >= perc90), 'ID']
```

A second approach for outlier detection is a posteriori measure comparing the treated and raw data. An interval is constructed around the perturbed values as described in the previous section. If the original values fall into the interval around the perturbed values, the perturbed values are considered unsafe since they are too close to the original values. There are different ways to construct such intervals, such as rank-based intervals and standard deviation-based intervals. *TeMe08* propose a robust alternative for these intervals. They construct the intervals based on the squared Robust Mahalanobis Distance (RMD) of the individual values. The intervals are scaled by the RMD such that outliers obtain larger intervals and hence need to have a larger perturbation in order to be considered safe than values that are not outliers. This method is implemented in *sdcMicro* in the function dRiskRMD(), which is an extension of the dRisk() function. This method is illustrated in the Section Case Studies.

5.8 Global risk

To construct one aggregate risk measure at the global level for the complete dataset, we can aggregate the measures for risk at the individual level in several ways. Global risk measures should be used with caution: behind an acceptable global risk can hide some very high-risk records that are compensated by many low risk records.

5.8.1 Mean of individual risk measures

A straightforward way of aggregating the individual risk measures is taking the mean of all individuals in the sample, which is equal to summing over all keys in the sample if multiplied by the sample frequencies of these keys and dividing by the sample size n:

$$R_1 = \frac{1}{n}\sum_i r_k = \frac{1}{n}\sum_k f_k r_k$$

r_k is the individual risk of key k that the i^{th} individual shares (see the Section *Categorical key variables and frequency counts*). This measure is reported as global risk in *sdcMicro*, is stored in the "risk" slot and can be displayed as shown in Listing 5.11. It indicates that the average re-identification probability is 0.01582 or 0.1582 %.

Listing 5.11: Computation of the global risk measure

```
# Global risk (average re-identification probability)
sdcInitial@risk$global$risk
[1] 0.01582
```

The global risk in the example data in Table 5.1 is 0.01582, which is the expected proportion of all individuals in the sample that could be re-identified by an intruder. Another way of expressing the global risk is the number of expected re-identifications, $n * R_1$, which is in the example 10 * 0.01582. The expected number of re-identifications is also saved in the *sdcMicro* object. Listing 5.12 shows how to display this.

Note: This global risk measure should be used with caution. The average risk can be relatively low, but a few individuals could have a very high probability of re-identification.

An easy way to check for this is to look at the distribution of the individual risk values or the number of individuals with risk values above a certain threshold, as shown in the next section.

Listing 5.12: Computation of expected number of re-identifications

```
# Global risk (expected number of reidentifications)
sdcInitial@risk$global$risk_ER
[1] 0.1582
```

5.8.2 Count of individuals with risks larger than a certain threshold

All individuals belonging to the same key have the same individual risk, r_k. Another way of expressing the total risk in the sample is the total number of observations that exceed a certain threshold of individual risk. Setting the threshold can be absolute (e.g., all those individuals who have a disclosure risk higher than 0.05 or 5%) or relative (e.g., all those individuals with risks higher than the upper quartile of individual risk). Listing 5.13 shows how, using *R*, one would count the number of observations with an individual re-identification risk higher than 5%. In the example, no individual has a higher disclosure risk than 0.05.

Listing 5.13: Number of individuals with individual risk higher than the threshold 0.05

```
sum(sdcInitial@risk$individual[,1] > 0.05)
[1] 0
```

These calculations can then be used to treat data for individuals whose risk values are above a predetermined threshold. We will see later that there are methods in *sdcMicro*, such as localSupp(), that can be used to suppress values of certain key variables for those individuals with risk above a specified threshold. This is explained further in the Section Local suppression .

5.9 Household risk

In many social surveys, the data have a hierarchical structure where an individual belongs to a higher-level entity (see the Section *Levels of risk*). Typical examples are households in social surveys or pupils in schools. Re-identification of one household member can lead to re-identification of the other household members, too. It is therefore easy to see that if we take the household structure into account, the re-identification risk is the risk that at least one of the

household members is re-identified.

$$r^h = P(A_1 \cup A_2 \cup \ldots \cup A_J) = 1 - \prod_{j=1}^{J} 1 - P(A_j),$$

where A_j is the event that the j^{th} member of the household is re-identified and $P(A_j) = r_k$ is the individual disclosure risk of the j^{th} member. For example, if a household member has three members with individual disclosure risks based on their respective keys 0.02, 0.03 and 0.03, respectively, the household risk is

$$1 - (1 - 0.02)(1 - 0.03)(1 - 0.03)) = 0.078$$

The hierarchical or household risk cannot be lower than the individual risk, and the household risk is always the same for all household members. The household risk should be used in cases where the data contain a hierarchical structure, i.e., where a household structure is present in the data. Using *sdcMicro*, if a household identifier is specified (in the argument *hhId* in the function createSdcObj()) while creating an *sdcMicro* object, the household risk will automatically be computed. Listing 5.14 shows how to display these risk measures.

Listing 5.14: Computation of household risk and expected number of re-identifications

```
# Household risk
sdcInitial@risk$global$hier_risk

# Household risk (expected number of reidentifications
sdcInitial@risk$global$hier_risk_ER
```

Note: The size of a household is an important identifier itself, especially for large households. Suppression of the actual size variable (e.g., number of household members), however, does not suffice to remove this information from the dataset, as a simple count of the household members for a particular household will allow reconstructing this variable as long as a household ID is in the data, which allows assigning individuals to households. We flag this for the reader's attention as it is important. Further discussion on approaches to the SDC process that take into account the household structure where it exists can be found in the Section Anonymization of the quasi-identifier household size

Recommended Reading Material on Risk Measurement

Elliot, Mark J, Anna Manning, Ken Mayes, John Gurd, and Michael Bane. 2005. "SUDA: A Program for Detecting Special Uniques." *Joint UNECE/Eurostat Work Session on Statistical Data Confidentiality*. Geneva.

Hundepool, Anco, Josep Domingo-Ferrer, Luisa Franconi, Sarah Giessing, Eric Schulte Nordholt, Keith Spicer, and Peter Paul de Wolf. 2012. *Statistical Disclosure Control.* Chichester: John Wiley & Sons Ltd. doi:10.1002/9781118348239.

Lambert, Diane. 1993."Measures of Disclosure Risk and Harm." *Journal of Official Statistics* 9(2) : 313-331.

Machanavajjhala, Ashwin, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. 2007. "L-diversity: Privacy Beyond K-anonymity." *ACM Trans. Knowl. Discov. Data* 1 (Article 3) (1556-4681). doi:10.1145/1217299.1217302. https://ptolemy.berkeley.edu/projects/truststc/pubs/465/L%20Diversity%20Privacy.pdf. Accessed July 6, 2018.

Templ, Matthias, Bernhard Meindl, Alexander Kowarik, and Shuang Chen. 2014. "Introduction to Statistical Disclosure Control (SDC)." http://www.ihsn.org/home/sites/default/files/resources/ihsn-working-paper-007-Oct27.pdf. August 1. Accessed July 6, 2018.

References

CHAPTER

SIX

ANONYMIZATION METHODS

This Section describes the SDC methods most commonly used. All methods are implementable in *R* by using the *sdcMicro* package. We discuss for every method for what type of data the method is suitable, both in terms of data characteristics and type of data. Furthermore, options such as specific parameters for each method are discussed as well as their impacts.[1] These findings are meant as guidance but should be used with caution, since every dataset has different characteristics and our findings may not always address your particular dataset. The last three sections are on the anonymization of variables and datasets with particular characteristics that deserve special attention. The Section *Anonymization of geospatial variables* deals with for anonymizing geographical data, such as GPS coordinates, the Section *Anonymization of the quasi-identifier household size* discusses the anonymization of data with a hierarchical structure (household structure) and the Section *Special case: census data* describes the peculiarities of dealing with and releasing census microdata.

To determine which anonymization methods are suitable for specific variables and/or datasets, we begin by presenting some classifications of SDC methods.

6.1 Classification of SDC methods

SDC methods can be classified as **non-perturbative** and **perturbative** (see *HDFG12*).

- **Non-perturbative methods** reduce the detail in the data by generalization or suppression of certain values (i.e., masking) without distorting the data structure.
- **Perturbative methods** do not suppress values in the dataset but perturb (i.e., alter) values to limit disclosure risk by creating uncertainty around the true values.

Both non-perturbative and perturbative methods can be used for categorical and continuous variables.

We also distinguish between **probabilistic** and **deterministic** SDC methods.

- **Probabilistic methods** depend on a probability mechanism or a random number-generating mechanism. Every time a probabilistic method is used, a different outcome is generated. For these methods it is often recommended that a seed be set for the random number generator if you want to produce replicable results.
- **Deterministic methods** follow a certain algorithm and produce the same results if applied repeatedly to the same data with the same set of parameters.

SDC methods for microdata intend to prevent identity and attribute disclosure. Different SDC methods are used for each type of disclosure control. Methods such as recoding and suppression are applied to quasi-identifiers to prevent identity disclosure, whereas top coding a quasi-identifier (e.g., income) or perturbing a sensitive variable prevent attribute disclosure.

As this practice guide is written around the use of the *sdcMicro* package, we discuss only SDC methods that are implemented in the *sdcMicro* package or can be easily implemented in *R*. These are the most commonly applied

[1] We also show code examples in *R,* which are drawn from findings we gathered by applying these methods to a large collection of datasets.

methods from the literature and used in most agencies experienced in using these methods. Table 6.1 gives an overview of the SDC methods discussed in this guide, their classification, types of data to which they are applicable and their function names in the *sdcMicro* package.

Table 6.1: SDC methods and corresponding functions in *sdcMicro*

Method	Classification of SDC method	Data Type	Function in sdcMicro
Global recoding	non-perturbative, determinitic	continuous and categorical	globalRecode , groupVars
Top and bottom coding	non-perturbative, determinitic	continuous and categorical	topBotCoding
Local suppression	non-perturbative, determinitic	categorical	localSuppression, localSupp
PRAM	perturbative, probabilistic	categorical	pram
Micro aggregation	perturbative, probabilistic	continuous	microaggregation
Noise addition	perturbative, probabilistic	continuous	addNoise
Shuffling	perturbative, probabilistic	continuous	shuffle
Rank swapping	perturbative, probabilistic	continuous	rankSwap

6.2 Non-perturbative methods

6.2.1 Recoding

Recoding is a deterministic method used to decrease the number of distinct categories or values for a variable. This is done by combining or grouping categories for categorical variables or constructing intervals for continuous variables. Recoding is applied to all observations of a certain variable and not only to those at risk of disclosure. There are two general types of recoding: global recoding and top and bottom coding.

Global recoding

Global recoding combines several categories of a categorical variable or constructs intervals for continuous variables. This reduces the number of categories available in the data and potentially the disclosure risk, especially for categories with few observations, but also, importantly, it reduces the level of detail of information available to the analyst. To illustrate recoding, we use the following example. Assume that we have five regions in our dataset. Some regions are very small and when combined with other key variables in the dataset, produce high re-identification risk for some individuals in those regions. One way to reduce risk would be to combine some of the regions by recoding them. We could, for example, make three groups out of the five, call them 'North', 'Central' and 'South' and re-label the values accordingly. This reduces the number of categories in the variable region from five to three.

Note: Any grouping should be some logical grouping and not a random joining of categories.

Examples would be grouping districts into provinces, municipalities into districts, or clean water categories together. Grouping all small regions without geographical proximity together is not necessarily the best option from the utility perspective. Table 6.2 illustrates this with a very simplified example dataset. Before recoding, three individuals have distinct keys, whereas after recoding (grouping 'Region 1' and 'Region 2' into 'North', 'Region 3' into 'Central' and 'Region 4' and 'Region 5' into 'South'), the number of distinct keys is reduced to four and the frequency of every key is at least two, based on the three selected quasi-identifiers. The frequency counts of the keys f_k are shown in the last column of Table 6.2. An intruder would find at least two individuals for each key and cannot distinguish any more

between individuals 1 – 3, individuals 4 and 6, individuals 5 and 7 and individuals 8 – 10, based on the selected key variables.

Table 6.2: Illustration of effect of recoding on frequency counts of keys

.	Before recoding				After recoding			
Individual	Region	Gender	Religion	f_k	Region	Gender	Religion	f_k
1	Region 1	Female	Catholic	1	North	Female	Catholic	3
2	Region 2	Female	Catholic	2	North	Female	Catholic	3
3	Region 2	Female	Catholic	2	North	Female	Catholic	3
4	Region 3	Female	Protestant	2	Central	Female	Protestant	2
5	Region 3	Male	Protestant	1	Central	Male	Protestant	2
6	Region 3	Female	Protestant	2	Central	Female	Protestant	2
7	Region 3	Male	Protestant	2	Central	Male	Protestant	2
8	Region 4	Male	Muslim	2	South	Male	Muslim	3
9	Region 4	Male	Muslim	2	South	Male	Muslim	3
10	Region 5	Male	Muslim	1	South	Male	Muslim	3

Recoding is commonly the first step in an anonymization process. It can be used to reduce the number of unique combinations of values of key variables. This generally increases the frequency counts for most keys and reduces the risk of disclosure. The reduction in the number of possible combinations is illustrated in Table 6.3 with the quasi-identifiers "region", "marital status" and "age". Table 6.3 shows the number of categories of each variable and the number of theoretically possible combinations, which is the product of the number of categories of each quasi-identifier, before and after recoding. "Age" is interpreted as a semi-continuous variable and treated as a categorical variable. The number of possible combinations and hence the risk for re-identification are reduced greatly by recoding. One should bear in mind that the number of possible combinations is a theoretical number; in practice, these may include very unlikely combinations such as age = 3 and marital status = widow and the actual number of combinations in a dataset may be lower.

Table 6.3: Illustration of the effect of recoding on the theoretically possible number of combinations an a dataset

Number of categories	Region	Marital status	Age	Possible combinations
before recoding	20	8	100	16,000
after recoding	6	6	15	540

The main parameters for global recoding are the size of the new groups, as well as defining which values are grouped together in new categories.

Note: Care should be taken to choose new categories in line with the data use of the end users and to minimize information loss as a result of recoding.

We illustrate this with three examples:

- *Age variable*: The categories of age should be chosen so that they still allow data users to make calculations relevant for the subject being studied. For example, if indicators need to be calculated for children of school going ages 6 – 11 and 12 – 17, and age needs to be grouped to reduce risk, then care should be taken to create age intervals that still allow the calculations to be made. A satisfactory grouping could be, for example, 0 – 5, 6 – 11, 12 – 17, etc., whereas a grouping 0 – 10, 11 – 15, 16 – 18 would destroy the data utility for these users. While it is common practice to create intervals (groups) of equal width (size), it is also possible (if data users require this) to recode only part of the variables and leave some values as they were originally. This could be done, for example, by recoding all ages above 20, but leaving those below 20 as they are. If SDC methods other than recoding will be used later or in a next step, then care should be taken when applying recoding to only part of the distribution, as this might increase the information loss due to the other methods, since the

grouping does not protect the ungrouped variables. Partial recoding followed by suppression methods such as local suppression may, for instance, lead to a higher number of suppressions than desired or necessary in case the recoding is done for the entire value range (see the next section on local suppression). In the example above, the number of suppressions of values below 20 will likely be higher than for values in the recoded range. The disproportionately high number of suppressions in this range of values that are not recoded can lead to higher utility loss for these groups.

- *Geographic variables*: If the original data specify administrative level information in detail, e.g., down to municipality level, then potentially those lower levels could be recoded or aggregated into higher administrative levels, e.g., province, to reduce risk. In doing so, the following should be noted: Grouping municipalities into abstract levels that intersect different provinces would make data analysis at the municipal or provincial level challenging. Care should be taken to understand what the user requires and the intention of the study. If a key component of the survey is to conduct analysis at the municipal level, then aggregating up to provincial level could damage the utility of the data for the user. Recoding should be applied if the level of detail in the data is not necessary for most data users and to avoid an extensive number of suppressions when using other SDC methods subsequently. If the users need information at a more detailed level, other methods such as perturbative methods might provide a better solution than recoding.
- *Toilet facility*: An example of a situation where a high level of detail might not be necessary and recoding may do very little harm to utility is the case of a detailed household toilet facility variable that lists responses for 20 types of toilets. Researchers may only need to distinguish between improved and unimproved toilet facilities and may not require the exact classification of up to 20 types. Detailed information of toilet types can be used to re-identify households, while recoding to two categories – improved and unimproved facilities – reduces the re-identification risk and in this context, hardly reduces data utility. This approach can be applied to any variable with many categories where data users are not interested in detail, but rather in some aggregate categories. Recoding addresses aggregation for the data users and at the same time protects the microdata. Important is to take stock of the aggregations used by data users.

Recoding should be applied only if removing the detailed information in the data will not harm most data users. If the users need information at a more detailed level, then recoding is not appropriate and other methods such as perturbative methods might work better.

In *sdcMicro* there are different options for global recoding. In the following paragraphs, we give examples of global recoding with the functions groupVars() and globalRecode(). The function groupVars() is generally used for categorical variables and the function globalRecode() for continuous variables. Finally, we discuss the use of rounding to reduce the detail in continuous variables.

Recoding a categorical variable using the sdcMicro function groupVars()

Assume that an object of class *sdcMicro* was created, which is called "sdcInitial"[2] (see the Section Objects of class sdcMicroObj) how to create objects of class *sdcMicro*). In Listing 6.1, the variable "sizeRes" has four different categories: 'capital, large city', 'small city', town', and 'countryside'). The first three are recoded or regrouped as 'urban' and the category 'countryside' is renamed 'rural'. In the function arguments, we specify the categories to be grouped (before) and the names of the categories after recoding (after). It is important that the vectors "before" and "after" have the same length. Therefore, we have to repeat 'urban' three times in the "after" vector to match the three different values that are recoded to 'urban'.

Note: The function groupVars() works only for variables of class factor.

We refer to the Section Classes in R on how to change the class of a variable.

[2] Here the *sdcMicro* object "sdcIntial" contains a dataset with 2,500 individuals and 103 variables. We selected five quasi-identifiers: "sizeRes", "age", "gender", "region", and "ethnicity".

Listing 6.1: Using the sdcMicro function groupVars() to recode a categorical variable

```
# Frequencies of sizeRes before recoding
table(sdcInitial@manipKeyVars$sizeRes)
## capital, large city          small city              town       countryside
##                 686                 310              146              1358

# Recode urban
sdcInitial  <-  groupVars(obj = sdcInitial, var = c("sizeRes"),
                          before = c("capital, large city", "small city", "town"),
                          after = c("urban", "urban", "urban"))

# Recode rural
sdcInitial  <-  groupVars(obj = sdcInitial, var = c("sizeRes"),
                          before = c("countryside"), after = c("rural"))

# Frequencies of sizeRes before recoding
table(sdcInitial@manipKeyVars$sizeRes)
## urban rural
##  1142  1358
```

Fig. 6.1 illustrates the effect of recoding the variable "sizeRes" and show respectively the frequency counts before and after recoding. We see that the number of categories has reduced from 4 to 2 and the small categories ('small city' and 'town') have disappeared.

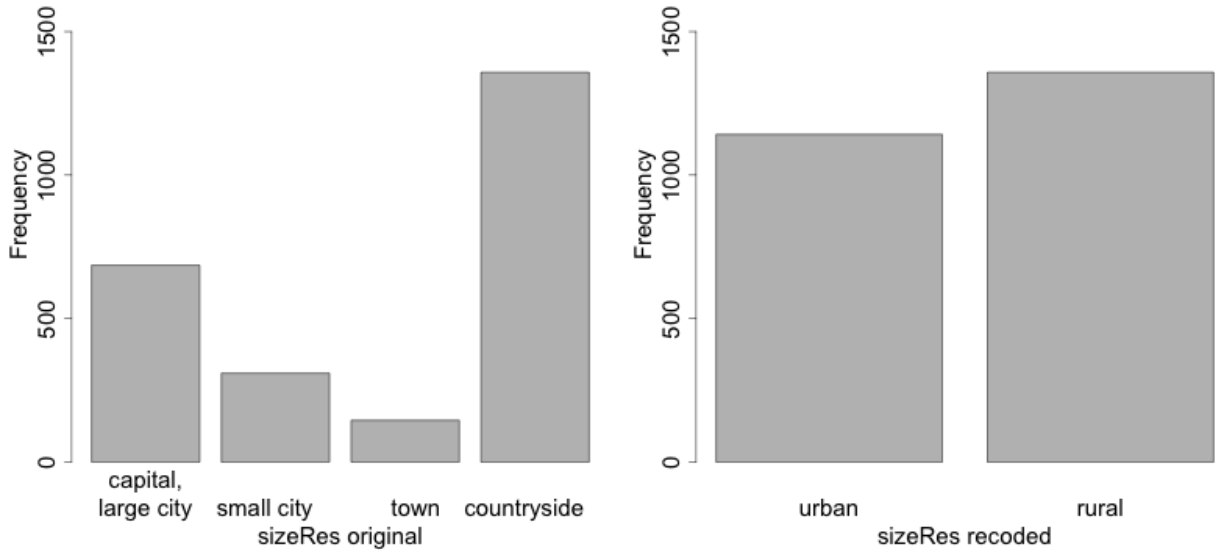

Fig. 6.1: Effect of recoding – frequency counts before and after recoding

Recoding a continuous variable using the sdcMicro function: globalRecode()

Global recoding of numerical (continuous) variables can be achieved in *sdcMicro* by using the function globalRecode(), which allows specifying a vector with the break points between the intervals. Recoding a continuous variable changes it into a categorical variable. One can additionally specify a vector of labels for the new categories. By de-

fault, the labels are the intervals, e.g., "(0, 10]". Listing 6.2 shows how to recode the variable age in 10-year intervals for age values between 0 and 100.

Note: Values that fall outside the specified intervals are assigned a missing value (NA).

Therefore, the intervals should cover the entire value range of the variable.

Listing 6.2: Using the *sdcMicro* function globalRecode() to recode a continuous variable (age)

```
sdcInitial <- globalRecode(sdcInitial, column = c('age'),
                           breaks = 10 * c(0:10))

# Frequencies of age after recoding
table(sdcInitial@manipKeyVars$age)
##    (0,10]   (10,20]   (20,30]   (30,40]   (40,50]   (50,60]   (60,70]   (70,80]   (80,90]
↪(90,100]
##       462       483       344       368       294       214       172        94        26
↪        3
```

Fig. 6.2 shows the effect of recoding the variable "age".

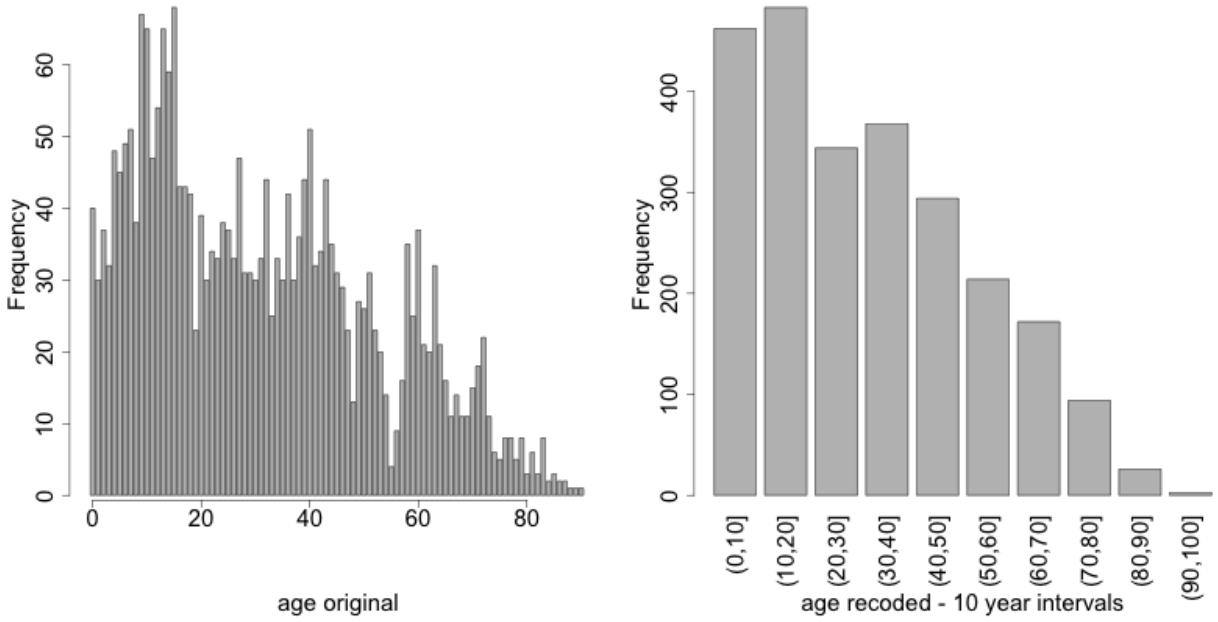

Fig. 6.2: Age variable before and after recoding

Instead of creating intervals of equal width, we can also create intervals of unequal width. This is illustrated in Listing 6.3, where we use the age groups 1-5, 6-11, 12-17, 18-21, 22-25, 26-49, 50-64 and 65+. In this example, this is a useful step, since even after recoding in 10-year intervals, the categories with high age values have low frequencies. We chose the intervals by respecting relevant school age and employment age values (e.g., retirement age is 65 in this example) such that the data can still be used for common research on education and employment. Fig. 6.3 shows the effect of recoding the variable "age".

Listing 6.3: Using globalRecode() to create intervals of unequal width

```
sdcInitial <- globalRecode(sdcInitial, column = c('age'),
                           breaks = c(0, 5, 11, 17, 21, 25, 49, 65, 100))

# Frequencies of age after recoding
table(sdcInitial@manipKeyVars$age)
##     (0,5]    (5,11]   (11,17]   (17,21]   (21,25]   (25,49]   (49,65] (65,100]
##       192       317       332       134       142       808       350       185
```

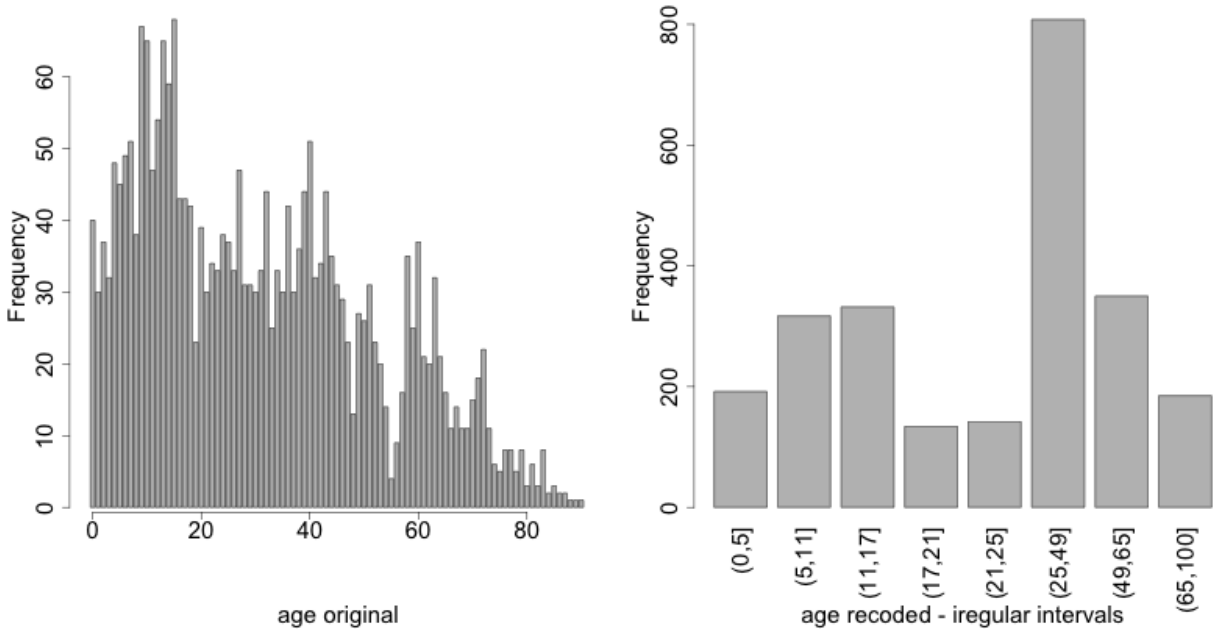

Fig. 6.3: Age variable before and after recoding

Caution about using the globalRecode() function in *sdcMicro*: In the current implementation of *sdcMicro*, the intervals are defined as **left-open**. In mathematical terms, this means that, in our example, age 0 is excluded from the specified intervals. In interval notation, this is denoted as (0, 5] (as in x-axis labels in Fig. 6.2 and Fig. 6.3 for the recoded variable). The interval (0, 5] is interpreted as from 0 to 5 and does not include 0, but does include 5. *R* recodes values that are not contained in any of the intervals as missing (NA). This implementation would set in our example all age values 0 (children under 1 year) to missing and could potentially mean a large data loss. The globalRecode() function allows only constructing intervals, which are left-open. This may not be a desirable result and the loss of the zero ages from the data is clearly problematic for a real-world dataset.

To construct **right-open** intervals, e.g., in our example, for age intervals [0,14), [15, 65), [66, 100), we present two alternatives for global recoding:

- A work-around for semi-continuous variables[3] that would allow for the globalRecode() to be used would be subtracting a small number from the boundary intervals, thus allowing the desired intervals to be created. In the following example, subtracting 0.1 from each interval forces globalRecode() to include 0 in the lowest interval and allow for breaks where we want them. We set the upper interval boundary to be larger than the maximum value for the "age" variable. We can use the option *labels* to define clear labels for the new categories. This is illustrated in Listing 6.4.

[3] This approach works only for semi-continuous variables, because in the case of continuous variables, there might be values that are between the lower interval boundary and the lower interval boundary minus the small number. For example, using this for income, we would have an interval (9999, 19999] and the value 9999.5 would be misclassified as belonging to the interval [10000, 19999].

Listing 6.4: Constructing right-open intervals for semi-continuous variables using built-in *sdcMicro* function globalRecode()

```
sdcInitial <- globalRecode(sdcInitial, column = c('age'),
                           breaks = c(-0.1, 14.9, 64.9, 99.9),
                           labels = c('[0,15)', '[15,65)', '[65,100)'))
```

- It is also possible to use *R* code to manually recode the variables without using *sdcMicro* functions. When using the built-in *sdcMicro* functions, the change in risk after recoding is automatically recalculated, but if recoded manually it is not. In this case, we need to take an extra step and recalculate the risk after manually changing the variables in the *sdcMicro* object. This approach is also valid for continuous variables and is illustrated in Listing 6.5.

Listing 6.5: Constructing intervals for semi-continuous and continuous variables using manual recoding in *R*

```
# Group age 0-14
sdcInitial@manipKeyVars$age[sdcInitial@manipKeyVars$age >= 0 &
sdcInitial@manipKeyVars$age < 15] <- 0

# Group age 15-64
sdcInitial@manipKeyVars$age[sdcInitial@manipKeyVars$age >= 15 &
sdcInitial@manipKeyVars$age < 65] <- 1

# Group age 65-100
sdcInitial@manipKeyVars$age[sdcInitial@manipKeyVars$age >= 65 &
sdcInitial@manipKeyVars$age <= 100] <- 2

# Add labels for the new values
sdcInitial@manipKeyVars$age <-ordered(sdcInitial@manipKeyVars$age,
levels = c(0,1,2), labels = c("0-14", "15-64", "65-100"))

# Recalculate risk after manual manipulation
sdcInitial <- calcRisks(sdcInitial)
```

Top and bottom coding

Top and bottom coding are similar to global recoding, but instead of recoding all values, only the top and/or bottom values of the distribution or categories are recoded. This can be applied only to ordinal categorical variables and (semi-)continuous variables, since the values have to be at least ordered. Top and bottom coding is especially useful if the bulk of the values lies in the center of the distribution with the peripheral categories having only few observations (outliers). Examples are age and income; for these variables, there will often be only a few observations above certain thresholds, typically at the tails of the distribution. The fewer the observations within a category, the higher the identification risk. One solution could be grouping the values at the tails of the distribution into one category. This reduces the risk for those observations, and, importantly, does so without reducing the data utility for the other observations in the distribution.

Deciding where to apply the threshold and what observations should be grouped requires:

- Reviewing the overall distribution of the variable to identify at which point the frequencies drop below the desired number of observations and identify outliers in the distribution. Fig. 6.4 shows the distribution of the age variable and suggests 65 (red vertical line) for the top code age.
- Taking into account the intended use of the data and the purpose for which the survey was conducted. For example, if the data are typically used to measure labor force participation for those aged 15 to 64, then top and bottom coding should not interfere with the categories 15 to 64. Otherwise the analyst would find it impossible

to create the desired measures for which the data were intended. In the example, we consider this and code all age higher than 64.

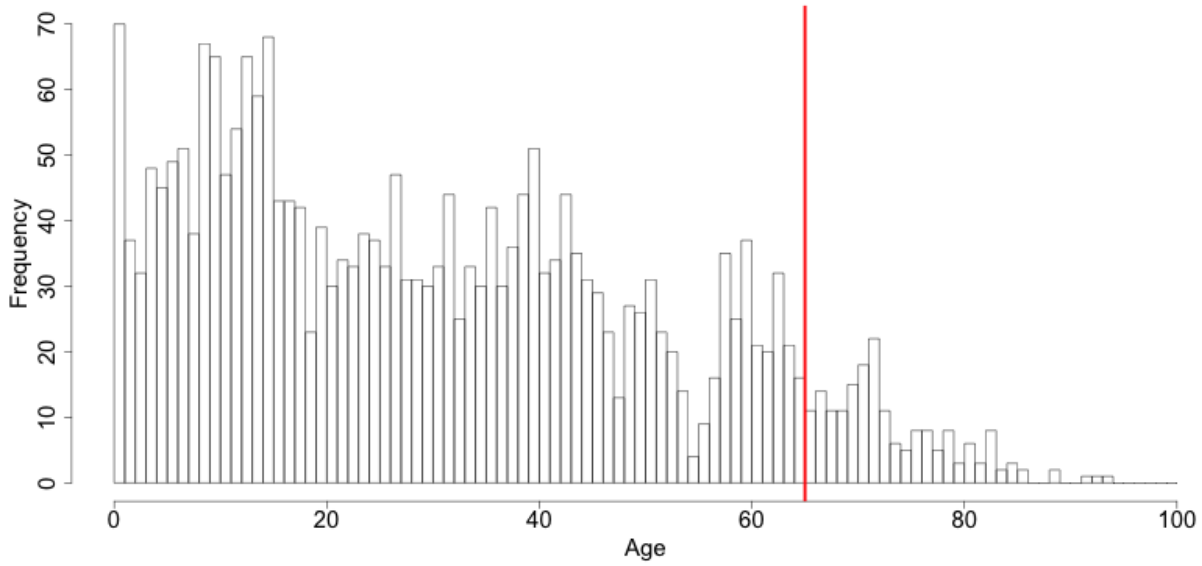

Fig. 6.4: Utilizing the frequency distribution of variable age to determine threshold for top coding

Top and bottom coding can be easily done with the function topBotCoding() in *sdcMicro*. Top coding and bottom coding cannot be done simultaneously in *sdcMicro*. Listing 6.6 illustrates how to recode values of age higher than 64 and values of age lower than 5; 65 and 5 replace the values respectively. To construct several top or bottom coding categories, e.g., age 65 – 80 and higher than age 80, one can use the groupVars() function in *sdcMicro* or manual recoding as described in the previous subsection.

Listing 6.6: Top coding and bottom coding in *sdcMicro* using topBotCoding() function

```
# Top coding at age 65
sdcInitial <- topBotCoding(obj = sdcInitial, value = 65, replacement = 65,
                           kind = 'top', column = 'age')

# Bottom coding at age 5
sdcInitial <- topBotCoding(obj = sdcInitial, value = 5, replacement = 5,
                           kind = 'bottom', column = 'age')
```

Rounding

Rounding is similar to grouping, but used for continuous variables. Rounding is useful to prevent exact matching with external data sources. In addition, it can be used to reduce the level of detail in the data. Examples are removing decimal figures or rounding to the nearest 1,000.

The next section discusses the method local suppression. Recoding is often used before local suppression to reduce the number of necessary suppressions.

Recommended Reading Material on Recoding

Hundepool, Anco, Josep Domingo-Ferrer, Luisa Franconi, Sarah Giessing, Rainer Lenz, Jane Naylor, Eric Schulte Nordholt, Giovanni Seri, and Peter Paul de Wolf. 2006. *Handbook on Statistical Disclosure Control.* ESSNet SDC. http://neon.vb.cbs.nl/casc/handbook.htm.

Hundepool, Anco, Josep Domingo-Ferrer, Luisa Franconi, Sarah Giessing, Eric Schulte Nordholt, Keith Spicer, and Peter Paul de Wolf. 2012. *Statistical Disclosure Control.* Chichester: John Wiley & Sons Ltd. doi:10.1002/9781118348239.

Templ, Matthias, Bernhard Meindl, Alexander Kowarik, and Shuang Chen. 2014. Statistical Disclosure Control (SDCMicro). http://www.ihsn.org/home/software/disclosure-control-toolbox. (accessed June 9, 2018).

De Waal, A.G., and Willenborg, L.C.R.J. 1999. *Information loss through global recoding and local suppression*. Netherlands Official Statistics, 14:17-20, 1999. Special issue on SDC

6.2.2 Local suppression

It is common in surveys to encounter values for certain variables or combinations of quasi-identifiers (keys) that are shared by very few individuals. When this occurs, the risk of re-identification for those respondents is higher than the rest of the respondents (see the Section k-anonymity). Often local suppression is used after reducing the number of keys in the data by recoding the appropriate variables. Recoding reduces the number of necessary suppressions as well as the computation time needed for suppression. Suppression of values means that values of a variable are replaced by a missing value (NA in *R*). The the Section k-anonymity discusses how missing values influence frequency counts and k-anonymity. It is important to note that not all values for all individuals of a certain variable are suppressed, which would be the case when removing a direct identifier, such as "name"; only certain values for a particular variable and a particular respondent or set of respondents are suppressed. This is illustrated in the following example and Table 6.4.

Table 6.4 presents a dataset with seven respondents and three quasi-identifiers. The combination {'female', 'rural', 'higher'} for the variables "gender", "region" and "education" is an unsafe combination, since it is unique in the sample. By suppressing either the value 'female' or 'higher', the respondent cannot be distinguished from the other respondents anymore, since that respondent shares the same combination of key variables with at least three other respondents. Only the value in the unsafe combination of the single respondent at risk is suppressed, not the values for the same variable of the other respondents. The freedom to choose which value to suppress can be used to minimize the total number of suppressions and hence the information loss. In addition, if one variable is very important to the user, we can choose not to suppress values of this variable, unless strictly necessary. In the example, we can choose between suppressing the value 'female' or 'higher' to achieve a safe data file; we chose to suppress 'higher'. This choice should be made taking into account the needs of data users. In this example we find "gender" more important than "education".

Table 6.4: Local suppression illustration - sample data before and after suppression

Variable	Before local suppression			After local suppression		
ID	Gender	Region	Education	Gender	Region	Education
1	female	rural	higher	female	rural	NA/missing[5]
2	male	rural	higher	male	rural	higher
3	male	rural	higher	male	rural	higher
4	male	rural	higher	male	rural	higher
5	female	rural	lower	female	rural	lower
6	female	rural	lower	female	rural	lower
7	female	rural	lower	female	rural	lower

Since continuous variables have a high number of unique values (e.g., income in dollars or age in years), k-anonymity and local suppression are not suitable for continuous variables or variables with a very high number of categories.

[5] In *R* suppressed values are recoded NA, the standard missing value code.

A possible solution in those cases might be to first recode to produce fewer categories (e.g., recoding age in 10-year intervals or income in quintiles). Always keep in mind, though, what effect any recoding will have on the utility of the data.

The *sdcMicro* package includes two functions for local suppression: localSuppression() and localSupp(). The function localSuppression() is most commonly used and allows the use of suppression on specified quasi-identifiers to achieve a certain level of k-anonymity for these quasi-identifiers. The algorithm used seeks to minimize the total number of suppressions while achieving the required k-anonymity threshold. By default, the algorithm is more likely to suppress values of variables with many different categories or values, and less likely to suppress variables with fewer categories. For example, the values of a geographical variable, with 12 different areas, are more likely to be suppressed than the values of the variable "gender", which has typically only two categories. If variables with many different values are important for data utility and suppression is not desired for them, it is possible to rank variables by importance in the localSuppression() function and thus specify the order in which the algorithm will seek to suppress values within quasi-identifiers to achieve k-anonymity. The algorithm seeks to apply fewer suppressions to variables of high importance than to variables with lower importance. Nevertheless, suppressions in the variables with high importance might be inevitable to achieve the required level of k-anonymity.

In Listing 6.7, local suppression is applied to achieve the k-anonymity threshold of 5 on the quasi-identifiers "gender", "region", "religion", "age" and "ethnicity"[6]. Without ranking the importance of the variables, the value of the variable "age" is more likely to be suppressed, since this is the variable with most categories. The variable "age" has 10 categories after recoding. The variable "gender" is least likely to be suppressed, since it has only two different values: 'male' and 'female'. The other variables have 4 ("sizeRes"), 2 ("region"), and 8 ("ethnicity") categories. After applying the localSuppression() function, we display the number of suppressions per variable with the built-in print() function with the option 'ls' for the local suppression output. As expected, the variable "age" has most suppressions (80). In fact, only the variable "ethnicity" of the other variables also needed suppressions (8) to achieve the k-anonymity threshold of 5. The variable "ethnicity" is the variable with the second highest number of suppressions. Subsequently, we undo and redo local suppression on the same data and reduce the number of suppressions on "age" by specifying the importance vector with high importance (little suppression) on the quasi-identifier "age". We also assign importance to the variable "gender". This is done by specifying an importance vector. The values in the importance vector can range from 1 to k, the number of quasi-identifiers. In our example k is equal to 5. Variables with lower values in the importance vectors have high importance and, when possible, receive fewer suppressions than variables with higher values.

To assign high importance to the variables "age" and "gender", we specify the importance vector as c(5, 1, 1, 5, 5), with the order according to the order of the specified variables in the *sdcMicro* object. The effect is clear: there are no suppressions in the variables "age" and "gender". For that, the other variables, especially "sizeRes" and "ethnicity", received many more suppressions. The total number of suppressed values has increased from 88 to 166.

Note: Fewer suppressions in one variable increase the number of necessary suppressions in other variables (cf. Listing 6.7).

Generally, the total number of suppressed values needed to achieve the required level of k-anonymity increases when specifying an importance vector, since the importance vector prevents to use the optimal suppression pattern. The importance vector should be specified only in cases where the variables with many categories play an important role in data utility for the data users[7].

Listing 6.7: Application of local suppression with and without importance vector

```
# local suppression without importance vector
sdcInitial <- localSuppression(sdcInitial, k = 5)
```

(continues on next page)

[6] Here the *sdcMicro* object "sdcIntial" contains a dataset with 2,500 individuals and 103 variables. We selected five quasi-identifiers: "sizeRes", "age", "gender", "region", and "ethnicity".

[7] This can be assessed with utility measures.

(continued from previous page)

```
print(sdcInitial, 'ls')
##     KeyVar | Suppressions (#) | Suppressions (%)
##    sizeRes |                0 |            0.000
##        age |               80 |            3.200
##     gender |                0 |            0.000
##     region |                0 |            0.000
##  ethnicity |                8 |            0.320

# Undoing the supressions
sdcInitial <- undolast(sdcInitial)

# Local suppression with importance vector to avoid suppressions
# in the first (gender) and fourth (age) variables
sdcInitial <- localSuppression(sdcInitial, importance = c(5, 1, 1, 5, 5), k = 5)
print(sdcInitial, 'ls')
##     KeyVar | Suppressions (#) | Suppressions (%)
##    sizeRes |               87 |            3.480
##        age |                0 |            0.000
##     gender |                0 |            0.000
##     region |               17 |            0.680
##  ethnicity |               62 |            2.480
```

Fig. 6.5 demonstrates the effect of the required k-anonymity threshold and the importance vector on the data utility by using several labor market-related indicators from an I2D2[8] dataset before and after anonymization. Fig. 6.5 displays the relative changes as a percentage of the initial value after re-computing the indicators with the data to which local suppression was applied. The indicators are the proportion of active females and males, and the number of females and males of working age. The values computed from the raw data were, respectively, 68%, 12%, 8,943 and 9,702. The vertical line at 0 is the benchmark of no change. The numbers indicate the required k-anonymity threshold (3 or 5) and the colors indicate the importance vector: red (no symbol) is no importance vector, blue (with * symbol) is high importance on the variable with the employment status information and dark green (with + symbol) is high importance on the age variable.

A higher k-anonymity threshold leads to greater information loss (i.e., larger deviations from the original values of the indicators, the 5's are further away from the benchmark of no change than the corresponding 3's) caused by local suppression. Reducing the number of suppressions on the employment status variable by specifying an importance vector does not improve the indicators. Instead, reducing the number of suppressions on age greatly reduces the information loss. Since specific age groups have a large influence on the computation of these indicators (the rare cases are in the extremes and will be suppressed), high suppression rates on age distort the indicators. It is generally useful to compare utility measures (see the Section Measuring Utility and Information Loss) to specify the importance vector, since the effects can be unpredictable.

The threshold of k-anonymity to be set depends on several factors, which are amongst others: 1) the legal requirements for a safe data file; 2) other methods that will be applied to the data; 3) the number of suppressions and related information loss resulting from higher thresholds; 4) the type of variable; 5) the sample weights and sample size; and 6) the release type (see the Section Release Types). Commonly applied levels for the k-anonymity threshold are 3 and 5.

Table 6.5 illustrates the influence of the importance vector and k-anonymity threshold on the running time, global risk after suppression and total number of suppressions required to achieve this k-anonymity threshold. The dataset contains about 63,000 individuals. The higher the k-anonymity threshold, the more suppressions are needed and the lower the risk after local suppression (expected number of re-identifications). In this particular example, the computation time is shorter for higher thresholds. This is due the higher number of necessary suppressions, which reduces the difficulty of the search for an optimal suppression pattern.

[8] I2D2 is a dataset with data related to the labor market.

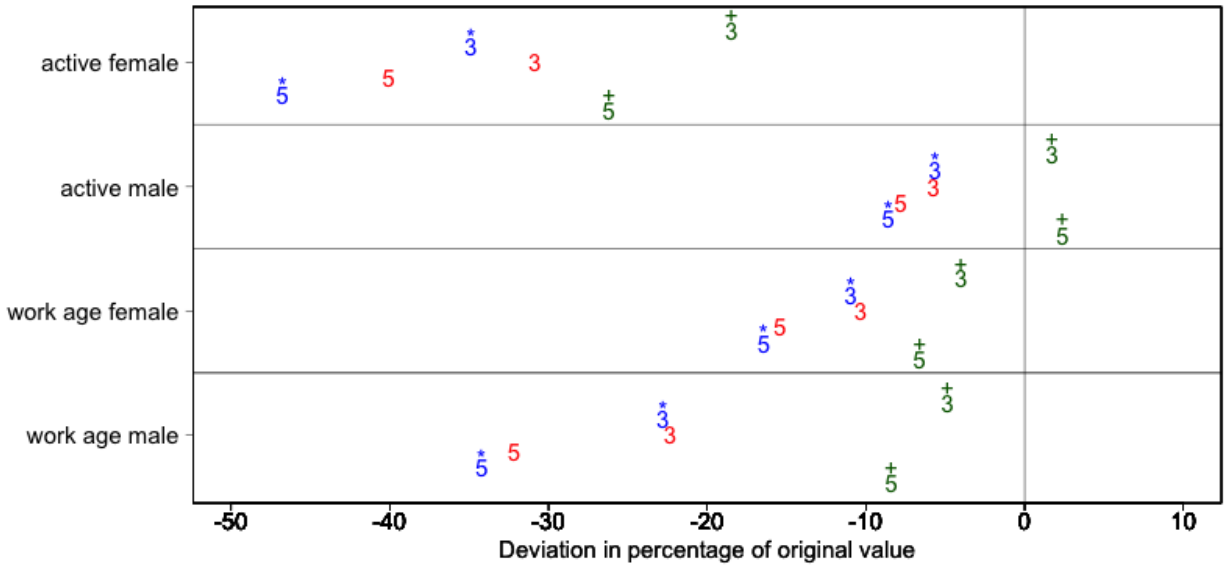

Fig. 6.5: Changes in labor market indicators after anonymization of I2D2 data

The age variable is recoded in five-year intervals and has 20 age categories. This is the variable with the highest number of categories. Prioritizing the suppression of other variables leads to a higher total number of suppressions and a longer computation time.

Table 6.5: How importance vectors and k-anonymity thresholds affect running time and total number of suppressions

Threshold k-anonimity	Importance vector	Total number of suppressions	Threshold k-anonimity	Importance vector	Total number of suppressions
3	none (default)	6,676	5,387	293.0	11.8
3	employment status	7,254	5,512	356.5	13.1
3	age variable	8,175	60	224.6	4.5
5	none (default)	9,971	7,894	164.6	8.5
5	employment status	11,668	8,469	217.0	10.2
5	age variable	13,368	58	123.1	3.8

In cases where there are a large number of quasi-identifiers and the variables have many categories, the number of possible combinations increases rapidly (see k-anonymity). If the number of variables and categories is very large, the computation time of the localSuppression() algorithm can be very long (see the Section Computation time on computation time). Also, the algorithm may not reach a solution, or may come to a solution that will not meet the specified level of k-anonymity. Therefore, reducing the number of quasi-identifiers and/or categories before applying local suppression is recommended. This can be done by recoding variables or selecting some variables for other (perturbative) methods, such as PRAM. This is to ensure that the number of suppressions is limited and hence the loss of data is limited to only those values that pose most risk.

In some datasets, it might prove difficult to reduce the number of quasi-identifiers and even after reducing the number of categories by recoding, the local suppression algorithm takes a long time to compute the required suppressions. A solution in such cases can be the so-called 'all-m approach' (see *Wolf15*). The all-m approach consists of applying the local suppression algorithm as described above to all possible subsets of size m of the total set of quasi-identifiers. The advantage of this approach is that the partial problems are easier to solve and computation time will be slower. Caution should be applied since this method does not necessarily lead to k-anonymity in the complete set of quasi-identifiers.

There are two possibilities to reach the same level of protection: 1) to choose a higher threshold for k or 2) to re-apply the local suppression algorithm on the complete set of quasi-identifiers after using the all-m approach to achieve the required threshold. In the second case, the all-m approach leads to a shorter computation time at the cost of a higher total number of suppressions.

Note: The required level is not achieved automatically on the entire set of quasi-identifiers if the all-m approach is used.

Therefore, it is important to evaluate the risk measures carefully after using the all-m approach.

In *sdcMicro* the all-m approach is implemented in the 'combs' argument in the localSuppression() function. The value for m is specified in the 'combs' argument and can also take on several values. The subsets of different sizes are then used sequentially in the local suppression algorithm. For example if 'combs' is set to c(3,9), first all subsets of size 3 are considered and subsequently all subsets of size 9. Setting the last value in the combs argument to the total number of key variables guarantees the achievement of k-anonymity for the complete dataset. It is also possible to specify different values for k for each subset size in the 'k' argument. If we would want to achieve 5-anonimity on the subsets of size 3 and subsequently 3-anonimity on the subsets of size 9, we would set the 'k' argument to c(5,3). Listing 6.8 illustrates the use of the all-m approach in *sdcMicro*.

Listing 6.8: The all-m approach in sdcMicro

```
# Apply k-anonymity with threshold 5 to all subsets of two key variables and
# subsequently to the complete dataset
sdcInitial <- localSuppression(sdcInitial, k = 5, combs = c(2, 5))
# Apply k-anonymity with threshold 5 to all subsets of three key variables and
# subsequently with threshold 2 to the complete dataset
sdcInitial <- localSuppression(sdcInitial, k = c(3, 5), combs = c(5, 2))
```

Table 6.6 presents the results of using the all-m approach of a test dataset with 9 key variables and 4,000 records. The table shows the arguments 'k' and 'combs' of the localSuppression() function, the number of k-anonymity violators for different levels of k as well as the total number of suppressions. We observe that the different combinations do not always lead to the required level of k-anonimity. For example, when setting $k = 3$, and combs 3 and 7, there are still 15 records in the dataset (with a total of 9 quasi-identifiers) that violate 3-anonimity after local suppression. Due to the smaller sample size, the gains in running time are not yet apparent in this example, since the rerunning algorithm several times takes up time. A larger dataset would benefit more from the all-m approach, as the algorithm would take longer in the first place.

Table 6.6: Effect of the all-m approach on k-anonymity

Arguments		Number of violators for different levels of k-anonimity on complete set			Total number of suppressions	Running time (seconds)
k	combs	k = 2	k = 3	k = 5		
Before local suppression		2,464	3,324	3,877	0	0.00
3	.	0	0	1,766	2,264	17.08
5	.	0	0	0	3,318	10.57
3	3	2,226	3,202	3,819	3,873	13.39
3	3, 7	15	108	1,831	6,164	46.84
3	3, 9	0	0	1,794	5,982	31.38
3	5, 9	0	0	1,734	6,144	62.30
5	3	2,047	3,043	3,769	3,966	12.88
5	3, 7	0	6	86	7,112	46.57
5	3, 9	0	0	0	7,049	24.13
5	5, 9	0	0	0	7,129	54.76
5, 3	3, 7	11	108	1,859	6,140	45.60
5, 3	3, 9	0	0	1,766	2,264	30.07
5, 3	5, 9	0	0	0	3,318	51.25

Often the dataset contains variables that are related to the key variables used for local suppression. Examples are rural/urban to regions in case regions are completely rural or urban or variables that are only answered for specific categories (e.g., sector for those working, schooling related variables for certain age ranges). In those cases, the variables rural/urban or sector might not be quasi-identifiers themselves, but could allow the intruder to reconstruct suppressed values in the quasi-identifiers region or employment status. For example, if region 1 is completely urban, and all other regions are only semi-urban or rural, a suppression in the variable region for a record in region 1 can be simply reconstructed by the rural/urban variable. Therefore, it is useful to suppress the values corresponding to the suppressions in those linked variables. Listing 6.9 illustrates how to suppress the values in the variable "rururb" corresponding to the suppressions in the region variable. All values of "rururb", which correspond to a suppressed value (NA) in the variable "region" are suppressed (set to NA).

Listing 6.9: Manually suppressing values in linked variables

```
# Suppress values of rururb in file if region is suppressed
file[is.na(sdcInitial@manipKeyVars$region) &
     !is.na(sdcInitial@origData$region),'sizRes'] <- NA
```

Alternatively, the linked variables can be specified when creating the *sdcMicro* object. The linked variables are called ghost variables. Any suppression in the key variable will lead to a suppression in the variables linked to that key variable. Listing 6.10 shows how to specify the linkage between "region" and "rururb" with ghost variables.

Listing 6.10: Suppressing values in linked variables by specifying ghost variables

```
# Ghost (linked) variables are specified as a list of linkages
ghostVars <- list()

# Each linkage is a list, with the first element the key variable and
# the second element the linked variable(s)
ghostVars[[1]] <- list()
ghostVars[[1]][[1]] <- "region"
ghostVars[[1]][[2]] <- c("sizeRes")

```

(continues on next page)

(continued from previous page)

```
## Create the sdcMicroObj
sdcInitial <- createSdcObj(file, keyVars = keyVars, numVars = numVars,
                           weightVar = weight, ghostVars = ghostVars)

# The manipulated ghost variables are in the slot manipGhostVars
sdcInitial@manipGhostVars
```

The simpler alternative for the localSuppression() function in *sdcMicro* is the localSupp() function. The localSupp() function can be used to suppress values of certain key variables of individuals with risks above a certain threshold. In this case, all values of the specified variable for respondents with a risk higher than the specified threshold will be suppressed. The risk measure used is the individual risk (see the Section Individual risk). This is useful if one variable has sensitive values that should not be released for individuals with high risks of re-identification. What is considered high re-identification probability depends on legal requirements. In the following example, the values of the variable "education" are suppressed for all individuals whose individual risk is higher than 0.1, which is illustrated in Listing 6.11. For an overview of the individual risk values, it can be useful to look at the summary statistics of the individual risk values as well as the number of suppressions.

Listing 6.11: Application of built-in *sdcMicro* function localSupp()

```
# Summary statistics
summary(sdcInitial@risk$individual[,1])
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.05882 0.10000 0.14290 0.26480 0.33330 1.00000

# Number of individuals with individual risk higher than 0.1
sum(sdcInitial@risk$individual[,1] > 0.1)
## [1] 1863

# local suppression
sdcInitial <- localSupp(sdcInitial, threshold = 0.1, keyVar = 'education')
```

6.3 Perturbative methods

Perturbative methods do not suppress values in the dataset, but perturb (alter) values to limit disclosure risk by creating uncertainty around the true values. An intruder is uncertain whether a match between the microdata and an external file is correct or not. Most perturbative methods are based on the principle of matrix masking, i.e., the altered dataset Z is computed as

$$Z = AXB + C$$

where X is the original data, A is a matrix used to transform the records, B is a matrix to transform the variables and C is a matrix with additive noise.

Note: Risk measures based on frequency counts of keys are no longer valid after applying perturbative methods.

This can be seen in Table 6.7 , which displays the same data before and after swapping some values. The swapped values are in italics. Both before and after perturbing the data, all observations violate k-anonymity at the level 3 (i.e., each key does not appear more than twice in the dataset). Nevertheless, the risk of **correct** re-identification of the records is reduced and hence information contained in other (sensitive) variables possibly not disclosed. With a certain probability, a match of the microdata with an external data file will be wrong. For example, an intruder would find one individual with the combination {'male', 'urban', 'higher'}, which is a sample unique. However, this match is not correct, since the original dataset did not contain any individual with these characteristics and hence the matched

individual cannot be a correct match. The intruder cannot know with certainty whether the information disclosed from other variables for that record is correct.

Table 6.7: Sample data before and after perturbation

Variable	Original data			After perturbing the data		
ID	Gender	Region	Education	Gender	Region	Education
1	female	rural	higher	female	rural	higher
2	female	rural	higher	female	rural	*lower*
3	male	rural	lower	male	rural	lower
4	male	rural	lower	*female*	rural	lower
5	female	urban	lower	*male*	urban	*higher*
6	female	urban	lower	female	urban	lower

One advantage of perturbative methods is that the information loss is reduced, since no values will be suppressed, depending on the level of perturbation. One disadvantage is that data users might have the impression that the data was not anonymized before release and will be less willing to participate in future surveys. Therefore, there is a need for reporting both for internal and external use (see the Section Step 11: Audit and Reporting).

An alternative to perturbative methods is the generation of synthetic data files with the same characteristics as the original data files. Synthetic data files are not discussed in these guidelines. For more information and an overview of the use of synthetic data as SDC method, we refer to *Drec11* and Section 3.8 in *HDFG12*. We discuss here five perturbative methods: Post Randomization Method (PRAM), microaggregation, noise addition, shuffling and rank swapping.

6.3.1 PRAM (Post RAndomization Method)

PRAM is a perturbative method for categorical data. This method reclassifies the values of one or more variables, such that intruders that attempt to re-identify individuals in the data do so, but with positive probability, the re-identification made is with the wrong individual. This means that the intruder might be able to match several individuals between external files and the released data files, but cannot be sure whether these matches are to the correct individual.

PRAM is defined by the transition matrix P, which specifies the transition probabilities, i.e., the probability that a value of a certain variable stays unchanged or is changed to any of the other $k-1$ values. k is the number of categories or factor levels within the variable to be PRAMmed. For example, if the variable region has 10 different regions, k equals 10. In case of PRAM for a single variable, the transition matrix is size $k * k$. We illustrate PRAM with an example of the variable "region", which has three different values: 'capital', 'rural1' and 'rural2'. The transition matrix for applying PRAM to this variable is size 3*3:

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0.05 & 0.8 & 0.15 \\ 0.05 & 0.15 & 0.8 \end{bmatrix}$$

The values on the diagonal are the probabilities that a value in the corresponding category is not changed. The value 1 at position (1,1) in the matrix means that all values 'capital' stay 'capital'; this might be a useful decision, since most individuals live in the capital and no protection is needed. The value 0.8 at position (2,2) means that an individual with value 'rural1' will stay with probability 0.8 'rural1'. The values 0.05 and 0.15 in the second row of the matrix indicate that the value 'rural1' will be changed to 'capital' or 'rural2' with respectively probability 0.05 and 0.15. If in the initial file we had 5,000 individuals with value 'capital' and resp. 500 and 400 with values 'rural1' and 'rural2', we expect after applying PRAM to have 5,045 individuals with capital, 460 with rural1 and 395 with rural2[9]. The recoding is done independently for each individual. We see that the tabulation of the variable "region" yields different results before and after PRAM, which are shown in Table 6.8. The deviation from the expectation is due to the fact that PRAM is a probabilistic method, i.e., the results depend on a probability-generating mechanism; consequently, the results can differ every time we apply PRAM to the same variables of a dataset.

[9] The 5,045 is the expectation computed as 5,000 * 1 + 500 * 0.05 + 400 * 0.05.

Note: The number of changed values is larger than one might think when inspecting the tabulations in Table 6.8. Not all 5,000 individuals with value captial after PRAM had this value before PRAM and the 457 individuals in rural1 after PRAM are not all included in the 500 individuals before PRAM. The number of changes is larger than the differences in the tabulation (cf. transition matrix).

Given that the transition matrix is known to the end users, there are several ways to correct statistical analysis of the data for the distortions introduced by PRAM.

Table 6.8: Tabulation of variable "region" before and after PRAM

Value	Tabulation before PRAM	Tabulation after PRAM
capital	5,000	5,052
rural1	500	457
rural2	400	391

One way to guarantee consistency between the tabulations before and after PRAM is to choose the transition matrix so that, in expectation, the tabulations before and after applying PRAM are the same for all variables.[10] This method is called invariant PRAM and is implemented in *sdcMicro* in the function pram(). The method pram() determines the transition matrix that satisfies the requirements for invariant PRAM.

Note: Invariant does not guarantee that cross-tabulations of variables (unlike univariate tabulations) stay the same.

In Listing 6.12, we give an example of invariant PRAM using *sdcMicro*.[11] PRAM is a probabilistic method and the results can differ every time we apply PRAM to the same variables of a dataset. To overcome this and make the results reproducible, it is good practice to set a seed for the random number generator in *R*, so the same random numbers will be generated every time.[12] The number of changed records per variable is also shown.

Listing 6.12: Producing reproducible PRAM results by using set.seed()

```
# Set seed for random number generator
set.seed(123)

# Apply PRAM to all selected variables
sdcInitial <- pram(obj = sdcInitial)
## Number of changed observations:
## - - - - - - - - - - -
## ROOF != ROOF_pram : 75 (3.75%)
## TOILET != TOILET_pram : 200 (10%)
## WATER != WATER_pram : 111 (5.55%)
## ELECTCON != ELECTCON_pram : 99 (4.95%)
## FUELCOOK != FUELCOOK_pram : 152 (7.6%)
## OWNMOTORCYCLE != OWNMOTORCYCLE_pram : 42 (2.1%)
## CAR != CAR_pram : 168 (8.4%)
## TV != TV_pram : 170 (8.5%)
## LIVESTOCK != LIVESTOCK_pram : 52 (2.6%)
```

[10] This means that the vector with the tabulation of the absolute frequencies of the different categories in the original data is an eigenvector of the transition matrix that corresponds to the unit eigenvalue.

[11] In this example and the following examples in this section, the *sdcMicro* object "sdcIntial" contains a dataset with 2,000 individuals and 39 variables. We selected five categorical quasi-identifiers and 9 variables for PRAM: "ROOF", "TOILET", "WATER", "ELECTCON", "FUELCOOK", "OWNMOTORCYCLE", "CAR", "TV", and "LIVESTOCK". These PRAM variabels were selected according to the requirements of this particular dataset and for illustrative purposes.

[12] The PRAM method in *sdcMicro* sometimes produces the following error: Error in factor(xpramed, labels = lev) : invalid 'labels'; length 6 should be 1 or 5. Under some circumstances, changing the seed can solve this error.

Table 6.9 shows the tabulation of the variable after applying invariant PRAM. We can see that the deviations from the initial tabulations, which are in expectation 0, are smaller than with the transition matrix that does not fulfill the invariance property. The remaining deviations are due to the randomness.

Table 6.9: Tabulation of variable "region" before and after (invariant) PRAM

Value	Tabulation before PRAM	Tabulation after PRAM	Tabulation after invariant PRAM
capital	5,000	5,052	4,998
rural1	500	457	499
rural2	400	391	403

Table 6.10 presents the cross-tabulations with the variable gender. Before applying invariant PRAM, the share of males in the city is much higher than the share of females (about 60%). This property is not maintained after invariant PRAM (the shares of males and females in the city are roughly equal), although the univariate tabulations are maintained. One solution is to apply PRAM separately for the males and females in this example[13]. This can be done by specifying the strata argument in the pram() function in *sdcMicro* (see below).

Table 6.10: Cross-tabulation of variable "region" and variable "gender" before and after invariant PRAM

.	Tabulation before PRAM		Tabulation after invariant PRAM	
Value	male	female	male	female
capital	3,056	1,944	2,623	2,375
rural1	157	343	225	274
rural2	113	287	187	216

The pram() function in *sdcMicro* has several options.

Note: If no options are set and the PRAM method is applied to an sdcMicro object, all PRAM variables selected in the sdcMicro object are automatically used for PRAM and PRAM is applied within the selected strata (see the Section Objects of class sdcMicroObj on *sdcMicro* objects for more details).

Alternatively, PRAM can also be applied to variables that are not specified in the *sdcMicro* object as PRAM variables, such as key variables, which is shown in Listing 6.13. In that case, however, the risk measures that are automatically computed will not be correct anymore, since the variables are perturbed. Therefore, if during the SDC process PRAM will be applied to some key variables, it is recommended to create a new *sdcMicro* object where the variables to be PRAMmed are selected as PRAM variables in the function createSdcObj().

Listing 6.13: Selecting the variable "toilet" to apply PRAM

```
# Set seed for random number generator
set.seed(123)

# Apply PRAM only to the variable TOILET
sdcInitial <- pram(obj = sdcInitial, variables = c ("TOILET"))
## Number of changed observations:
## - - - - - - - - - - -
## TOILET != TOILET_pram : 115 (5.75%)
```

[13] This can also be achieved with multidimensional transition matrices. In that case, the probability is not specified for 'male' -> 'female', but for 'male' + 'rural' -> 'female' + 'rural' and for 'male' + 'urban' -> 'female' + 'urban'. This is not implemented in sdcMicro but can be achieved by PRAMming the males and females separately. In the example here, this could be done by specifying gender as strata variable in the pram() function in *sdcMicro*.

The results for PRAM differ if applied simultaneously to several variables or subsequently to each variable separately. It is not possible to specify the entire transition matrix in *sdcMicro*, but we can set minimum values (between 0 and 1) for the diagonal entries. The diagonal entries specify the probability that a certain value stays the same after applying PRAM. Setting the minimum value to 1 will yield no changes to this category. By default, this value is 0.8, which applies for all categories. It is also possible to specify a vector with value for each diagonal element of the transformation matrix/category. In Listing 6.14 values of the first region are less likely to change than values of the other regions.

Note: The invariant PRAM method requires that the transition matrix has a unit eigenvalue.

Not all sets of restrictions can therefore be used (e.g., the minimum value 1 on any of the categories).

Listing 6.14: Specifying minimum values for diagonal entries in PRAM transition matrix

```
sdcInitial <- pram(obj = sdcInitial, variables = c("TOILET"),
                   pd = c(0.9, 0.5, 0.5, 0.5))
## Number of changed observations:
## - - - - - - - - - - -
## TOILET != TOILET_pram : 496 (24.8%)
```

In the invariant PRAM method, we can also specify the amount of perturbation by specifying the parameter alpha. This choice is reflected in the transition matrix. By default, the alpha value is 0.5. The larger alpha, the larger the perturbations. Alpha equal to zero leads to no changes. The maximum value for alpha is 1.

PRAM is especially useful when a dataset contains many variables and applying other anonymization methods, such as recoding and local suppression, would lead to significant information loss. Checks on risk and utility are important after PRAM.

To do statistical inference on variables to which PRAM was applied, the researcher needs knowledge about the PRAM method as well as about the transition matrix. The transition matrix, together with the random number seed, can, however, lead to disclosure through reconstruction of the non-perturbed values. Therefore, publishing the transition matrix but not the random seed is recommended.

A disadvantage of using PRAM is that very unlikely combinations can be generated, such as a 63-year-old who goes to school. Therefore, the PRAMmed variables need to be audited to prevent such combinations from happening in the released data file. In principal, the transition matrix can be designed in such a way that certain transitions are not possible (probability 0). For instance, for those that go to school, the age must range within 6 to 18 years and only such changes are allowed. In *sdcMicro* the transition matrix cannot be exactly specified. A useful alternative is constructing strata and applying PRAM within the strata. In this way, the changes between variables will only be applied within the strata. Listing 6.15 illustrates this by applying PRAM to the variable "toilet" within the strata generated by the "region" education. This prevents changes in the variable "toilet", where toilet types in a particular region are exchanged with those in other regions. For instance, in the capital region certain types of unimproved toilet types are not in use and therefore these combinations should not occur after PRAMming. Values are only changed with those that are available in the same strata. Strata can be formed by any categorical variable, e.g., gender, age groups, education level.

Listing 6.15: Minimizing unlikely combinations by applying PRAM within strata

```
# Applying PRAM within the strata generated by the variable region
sdcInitial <- pram(obj = sdcInitial, variables = c("TOILET"),
                   strata_variables = c("REGION"))
## Number of changed observations:
## - - - - - - - - - - -
## TOILET != TOILET_pram : 179 (8.95%)
```

Recommended Reading Material on PRAM

Gouweleeuw, J. M, P Kooiman, L.C.R.J Willenborg, and P.P de Wolf. "Post Randomization for Statistical Disclosure Control: Theory and Implementation." *Journal of Official Statistics* 14, no. 4 (1998a): 463-478. Available at http://www.jos.nu/articles/abstract.asp?article=144463

Gouweleeuw, J. M, P Kooiman, L.C.R.J Willenborg, and Peter Paul de Wolf. "The Post Randomization Method for Protecting Microdata." *Qüestiió, Quaderns d'Estadística i Investigació Operativa 22,* no. 1 (1998b): 145-156. Available at http://www.raco.cat/index.php/Questiio/issue/view/2250

Marés, Jordi, and Vicenç Torra. 2010."PRAM Optimization Using an Evolutionary Algorithm." *In Privacy in Statistical Databases*, by Josep Domingo-Ferrer and Emmanouil Magkos, 97-106. Corfú, Greece: Springer.

Warner, S.L. "Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias." *Journal of American Statistical Association* 57 (1965): 622-627.

6.3.2 Microaggregation

Microaggregation is most suitable for continuous variables, but can be extended in some cases to categorical variables.[14] It is most useful where confidentiality rules have been predetermined (e.g., a certain threshold for k-anonymity has been set) that permit the release of data only if combinations of variables are shared by more than a predetermined threshold number of respondents (k). The first step in microaggregation is the formation of small groups of individuals that are homogeneous with respect to the values of selected variables, such as groups with similar income or age. Subsequently, the values of the selected variables of all group members are replaced with a common value, e.g., the mean of that group. Microaggregation methods differ with respect to (i) how the homogeneity of groups is defined, (ii) the algorithms used to find homogeneous groups, and (iii) the determination of replacement values. In practice, microaggregation works best when the values of the variables in the groups are more homogeneous. When this is the case, then the information loss due to replacing values with common values for the group will be smaller than in cases where groups are less homogeneous.

In the univariate case, and also for ordinal categorical variables, formation of homogeneous groups is straightforward: groups are formed by first ordering the values of the variable and then creating g groups of size n_i for all groups i in $1, \ldots, g$. This maximizes the within-group homogeneity, which is measured by the within-groups sum of squares (SSE)

$$SSE = \sum_{i=1}^{g} \sum_{j=1}^{n_i} (x_{ij} - \overline{x}_i)^T (x_{ij} - \overline{x}_i)$$

The lower the SSE, the higher the within-group homogeneity. The group sizes can differ amongst groups, but often groups of equal size are used to simplify the search[15].

The function microaggregation() in *sdcMicro* can be used for univariate microaggregation. The argument 'aggr' specifies the group size. Forming groups is easier if all groups – except maybe the last group of remainders – have the same size. This is the case in the implementation in *sdcMicro* as it is not possible to have groups of different sizes. Listing 6.16 shows how to use the function microaggregation() in *sdcMicro*.[16] The default group size is 3 but the user can specify any desired group size. Choice of group size depends on the homogeneity within the groups and the required level of protection. In general it holds that the larger the group, the higher the protection. A disadvantage of groups of equal sizes is that the data might be unsuitable for this. For instance, if two individuals have a low income (e.g., 832 and 966) and four individuals have a high income (e.g., 3,313, 3,211, 2,987, 3,088), the mean of two groups

[14] Microaggregation can also be used for categorical data, as long as there is a possibility to form groups and an aggregate replacement for the values in the group can be calculated. This is the case for ordinal variables.

[15] Here all groups can have different sizes (i.e., number of individuals in a group). In practice, the search for homogeneous groups is simplified by imposing equal group sizes for all groups.

[16] In this example and the following examples in this section, the *sdcMicro* object "sdcIntial" contains a dataset with 2,000 individuals and 39 variables. We selected five categorical quasi-identifiers and three continuous quasi-identifiers: "INC", "EXP" and "WEALTH".

of size three (e.g., (832 + 966 + 2,987) / 3 = 1,595 and (3,088 + 3,211 + 3,313) / 3 = 3,204) would represent neither the low nor the high income.

Listing 6.16: Applying univariate microaggregation with *sdcMicro* function microaggregation()

```
sdcInitial <- microaggregation(obj = sdcInitial, variables = 'INC',
                               aggr = 3, method = mafast, measure = "mean")
```

By default, the microaggregation function replaces values with the group mean. An alternative, more robust approach is to replace group values with the median. This can be specified in the argument 'measure' of the function microaggregation(). In cases where the median is chosen, one individual in every group keeps the same value if groups have odd sizes. In cases where there is a high degree of heterogeneity within the groups (this is often the case for larger groups), the median is preferred to preserve the information in the data. An example is income, where one outlier can lead to multiple outliers being created when using microaggregation. This is illustrated in Table 6.11. If we choose the mean as replacement for all values, which are grouped with the outlier (6,045 in group 2), these records will be assigned values far from their original values. If we chose the median, the incomes of individuals 1 and 2 are not perturbed, but no value is an outlier. Of course, this might in itself present problems.

Note: If microaggregation alters outlying values, this can have a significant impact on the computation of some measures sensitive to outliers, such as the GINI index.

In the case where microaggregation is applied to categorical variables, the median is used to calculate the replacement value for the group.

Table 6.11: Illustrating the effect of choosing mean vs. median for microaggregation where outliers are concerned

ID	Group	Income	Microaggregation (mean)	Microaggregation (median)
1	1	2,300	2,245	2,300
2	2	2,434	3,608	2,434
3	1	2,123	2,245	2,300
4	1	2,312	2,245	2,300
5	2	6,045	3,608	2,434
6	2	2,345	3,608	2,434

In case of multiple variables that are candidates for microaggregation, one possibility is to apply univariate microaggregation to each of the variables separately. The advantage of univariate microaggregation is minimal information loss, since the changes in the variables are limited. The literature shows, however, that disclosure risk can be very high if univariate microaggregation is applied to several variables separately and no additional anonymization techniques are applied (*DMOT02*). To overcome this shortcoming, an alternative to univariate microaggregation is multivariate microaggregation.

Multivariate microaggregation is widely used in official statistics. The first step in multivariate aggregation is the creation of homogeneous groups based on several variables. Groups are formed based on multivariate distances between the individuals. Subsequently, the values of all variables for all group members are replaced with the same values. Table 6.12 illustrates this with three variables. We see that the grouping by income, expenditure and wealth leads to a different grouping, as in the case in Table 6.11, where groups were formed based only on income.

Table 6.12: Illustration of multivariate microaggregation

ID	Group	Before microaggregation			After microaggregation		
.	.	Income	Exp	Wealth	Income	Exp	Wealth
1	1	2,300	1,714	5.3	2,285.7	1,846.3	6.3
2	1	2,434	1,947	7.4	2,285.7	1,846.3	6.3
3	1	2,123	1,878	6.3	2,285.7	1,846.3	6.3
4	2	2,312	1,950	8.0	3,567.3	2,814.0	8.3
5	2	6,045	4,569	9.2	3,567.3	2,814.0	8.3
6	2	2,345	1,923	7.8	3,567.3	2,814.0	8.3

There are several multivariate microaggregation methods that differ with respect to the algorithm used for creating groups of individuals. There is a trade-off between speed of the algorithm and within-group homogeneity, which is directly related to information loss. For large datasets, this is especially challenging. We discuss the Maximum Distance to Average Vector (MDAV) algorithm here in more detail. The MDAV algorithm was first introduced by *DoTo05* and represents a good choice with respect to the trade-off between computation time and the group homogeneity, computed by the within-group SSE. The MDAV algorithm is implemented in *sdcMicro*.

The algorithm computes an average record or centroid C, which contains the average values of all included variables. We select an individual A with the largest squared Euclidean distance from C, and build a group of k records around A. The group of k records is made up of A and the $k-1$ records closest to A measured by the Euclidean distance. Next, we select another individual B, with the largest squared Euclidean distance from individual A. With the remaining records, we build a group of k records around B. In the same manner, we select an individual D with the largest distance from B and, with the remaining records, build a new group of k records around D. The process is repeated until we have fewer than $2*k$ records remaining. The MDAV algorithm creates groups of equal size with the exception of maybe one last group of remainders. The microaggregated dataset is then computed by replacing each record in the original dataset by the average values of the group to which it belongs. Equal group sizes, however, may not be ideal for data characterized by greater variability. In *sdcMicro* multivariate microaggregation is also implemented in the function microaggregation(). Listing 6.17 shows how to choose the MDAV algorithm in *sdcMicro*.

Listing 6.17: Multivariate microaggregation with the Maximum Distance to Average Vector (MDAV) algorithm in *sdcMicro*

```
sdcInitial <- microaggregation(obj = sdcInitial,
                               variables = c("INC", "EXP", "WEALTH"),
                               method = "mdav")
```

It is also possible to group variables only within strata. This reduces the computation time and adds an extra layer of protection to the data, because of the greater uncertainty produced[17]. In *sdcMicro* this can be achieved by specifying the strata variables, as shown in Listing 6.18.

Listing 6.18: Specifying strata variables for microaggregation

```
sdcInitial <- microaggregation(obj = sdcInitial,
                               variables = c("INC", "EXP", "WEALTH"),
                               method = "mdav", strata_variables = c("strata"))
```

Besides the method MDAV, there are few other grouping methods implemented in *sdcMicro* (*TeMK14*). Table 6.13 gives an overview of these methods. Whereas the method 'MDAV' uses the Euclidian distance, the method 'rmd' uses the Mahalanobis distance instead. An alternative to these methods is sorting the respondents based on the first principal component (PC), which is the projection of all variables into a one-dimensional space maximizing the variance of this projection. The performance of this method depends on the share of the total variance in the data that is explained by the first PC. The 'rmd' method is computationally more intensive due to the computation of Mahalanobis distances,

[17] Also the homogeneity in the groups will be generally lower, leading to larger changes, higher protection, but also more information loss, unless the strata variable correlates with the microaggregation variable.

but provides better results with respect to group homogeneity. It is recommended for smaller datasets (*TeMK14*).

Table 6.13: Grouping methods for microaggregation that are implemented in *sdcMicro*

Method / option in sdcMicro	Description
mdav	grouping is based on classical (Euclidean) distance measures
rmd	grouping is based on robust multivariate (Mahalanobis) distance measures
pca	grouping is based on principal component analysis whereas the data are sorted on the first principal component
clustpppca	grouping is based on clustering and (robust) principal component analysis for each cluster
influence	grouping is based on clustering and aggregation is performed within clusters

In case of several variables to be used for microaggregation, looking first at the covariance or correlation matrix of these variables is recommended. If not all variables correlate well, but two or more sets of variables show high correlation, less information loss will occur when applying microaggregation separately to these sets of variables. In general, less information loss will occur when applying multivariate microaggregation, if the variables are highly correlated. The advantage of replacing the values with the mean of the groups rather than other replacement values has the advantage that the overall means of the variables are preserved.

Recommended Reading Material on Microaggregation

Domingo-Ferrer, Josep, and Josep Maria Mateo-Sanz. 2002."Practical data-oriented microaggregation for statistical disclosure control." *IEEE Transactions on Knowledge and Data Engineering 14* (2002): 189-201.

Hansen, Stephen Lee, and Sumitra Mukherjee. 2003. "A polynomial algorithm for univariate optimal." *IEEE Transactions on Knowledge and Data Engineering* 15 (2003): 1043-1044.

Hundepool, Anco, Josep Domingo-Ferrer, Luisa Franconi, Sarah Giessing, Rainer Lenz, Jane Naylor, Eric Schulte Nordholt, Giovanni Seri, and Peter Paul de Wolf. 2006. *Handbook on Statistical Disclosure Control.* ESSNet SDC. http://neon.vb.cbs.nl/casc/handbook.htm

Hundepool, Anco, Josep Domingo-Ferrer, Luisa Franconi, Sarah Giessing, Eric Schulte Nordholt, Keith Spicer, and Peter Paul de Wolf. 2012. *Statistical Disclosure Control.* Chichester: John Wiley & Sons Ltd. doi:10.1002/9781118348239.

Templ, Matthias, Bernhard Meindl, Alexander Kowarik, and Shuang Chen. 2014, August. "International Household Survey Network (IHSN)." http://www.ihsn.org/home/software/disclosure-control-toolbox. (accessed July 9, 2018).

6.3.3 Noise addition

Noise addition, or noise masking, means adding or subtracting (small) values to the original values of a variable, and is most suited to protect continuous variables (see *Bran02* for an overview). Noise addition can prevent exact matching of continuous variables. The advantages of noise addition are that the noise is typically continuous with mean zero, and exact matching with external files will not be possible. Depending on the magnitude of noise added, however, approximate interval matching might still be possible.

When using noise addition to protect data, it is important to consider the type of data, the intended use of the data and the properties of the data before and after noise addition, i.e., the distribution – particularly the mean – covariance and correlation between the perturbed and original datasets.

Depending on the data, it may also be useful to check that the perturbed values fall within a meaningful range of values. Fig. 6.7 illustrates the changes in data distribution with increasing levels of noise. For data that has outliers,

it is important to note that when the perturbed data distribution is similar to the original data distribution (e.g., at low noise levels), noise addition will not protect outliers. After noise addition, these outliers can generally still be detected as outliers and hence easily be identified. An example is a single very high income in a certain region. After perturbing this income value, the value will still be recognized as the highest income in that region and can thus be used for re-identification. This is illustrated in Fig. 6.6, where 10 original observations (open circles) and the anonymized observations (red triangles) are plotted. The tenth observation is an outlier. The values of the first nine observations are sufficiently protected by adding noise: their magnitude and order has changed and exact or interval matching can be successfully prevented. The outlier is not sufficiently protected since, after noise addition, the outlier can still be easily identified. The fact that the absolute value has changed is not sufficient protection. On the other hand, at high noise levels, protection is higher even for the outliers, but the data structure is not preserved and the information loss is large, which is not an ideal situation. One way to circumvent the outlier problem is to add noise of larger magnitude to outliers than to the other values.

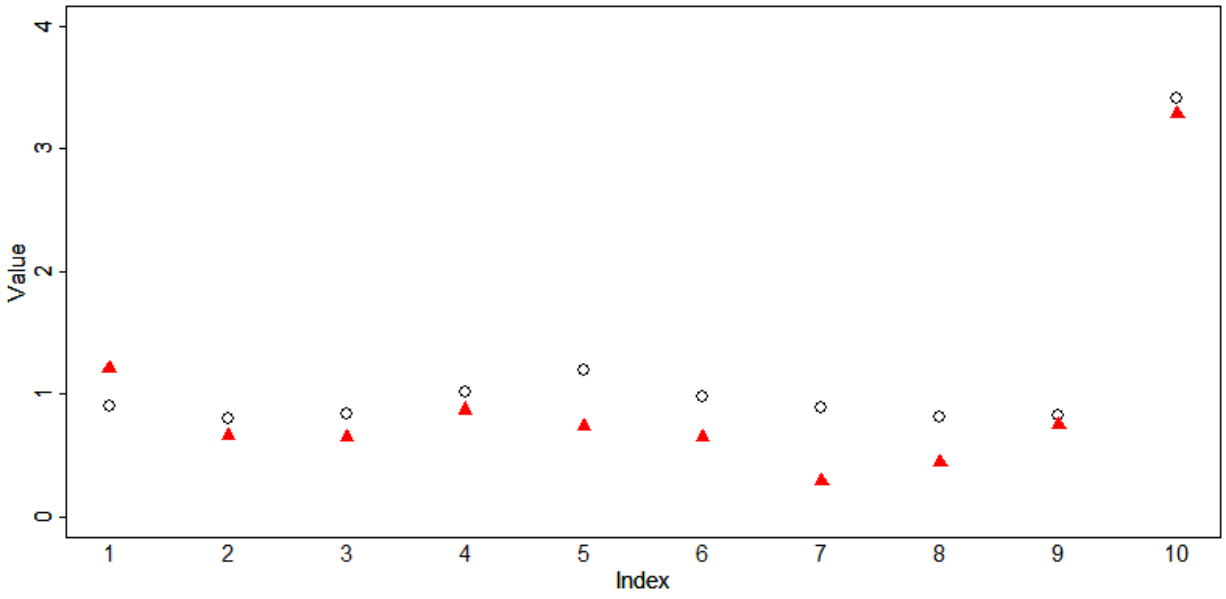

Fig. 6.6: Illustration of effect of noise addition to outliers

There are several noise addition algorithms. The simplest version of noise addition is uncorrelated additive normally distributed noise, where x_j, the original values of variable $j$are replaced by

$$z_j = x_j + \varepsilon_j,$$

where $\varepsilon_j \sim N(0, \sigma^2_{\varepsilon_j})$ and $\sigma_{\varepsilon_j} = \alpha * \sigma_j$ with σ_j the standard deviation of the original data. In this way, the mean and the covariances are preserved, but not the variances and correlation coefficient. If the level of noise added, α, is disclosed to the user, many statistics can be consistently estimated from the perturbed data. The added noise is proportional to the variance of the original variable. The magnitude of the noise added is specified by the parameter α, which specifies this proportion. The standard deviation of the perturbed data is $1 + \alpha$ times the standard deviation of the perturbed data. A decision on the magnitude of noise added should be informed by the legal situation regarding data privacy, data sensitivity and the acceptable levels of disclosure risk and information loss. In general, the level of noise is a function of the variance of the original variables, the level of protection needed and the desired value range after anonymization[18]. An α value that is too small will lead to insufficient protection, while an α value that is too high will make the data useless for data users.

In *sdcMicro* noise addition is implemented in the function addNoise(). The algorithm and parameter can be specified as arguments in the function addNoise(). Simple noise addition is implemented in the function addNoise() with the value

[18] Common values for α are between 0.5 and 2. The default value in the *sdcMicro* function addNoise() is 150, which is too large for most datasets; the level of noise should be set in the argument 'noise'.

"additive" for the argument 'method'. Listing 6.19 shows how to use *sdcMicro* to add uncorrelated noise to expenditure variables, where the standard deviation of the added noise equals half the standard deviation of the original variables.[19] Noise is added to all selected variables.

Listing 6.19: Uncorrelated noise addition

```
sdcInitial <- addNoise(obj = sdcInitial,
                       variables = c('TOTFOOD', 'TOTHLTH', 'TOTALCH', 'TOTCLTH',
                                     'TOTHOUS', 'TOTFURN', 'TOTTRSP', 'TOTCMNQ',
                                     'TOTRCRE', 'TOTEDUC', 'TOTHOTL', 'TOTMISC'),
                       noise = 0.5, method = "additive")
```

Fig. 6.7 shows the frequency distribution of a numeric continuous variable and the distribution before and after noise addition with different levels of noise (0.1, 0.5, 1, 2 and 5). The first plot shows the distribution of the original values. The histograms clearly show that noise of large magnitudes (high values of alpha) lead to a distribution of the data far from the original values. The distribution of the data changes to a normal distribution when the magnitude of the noise grows respective to the variance of the data. The mean in the data is preserved, but, with an increased level of noise, the variance of the perturbed data grows. After adding noise of magnitude 5, the distribution of the original data is completely destroyed.

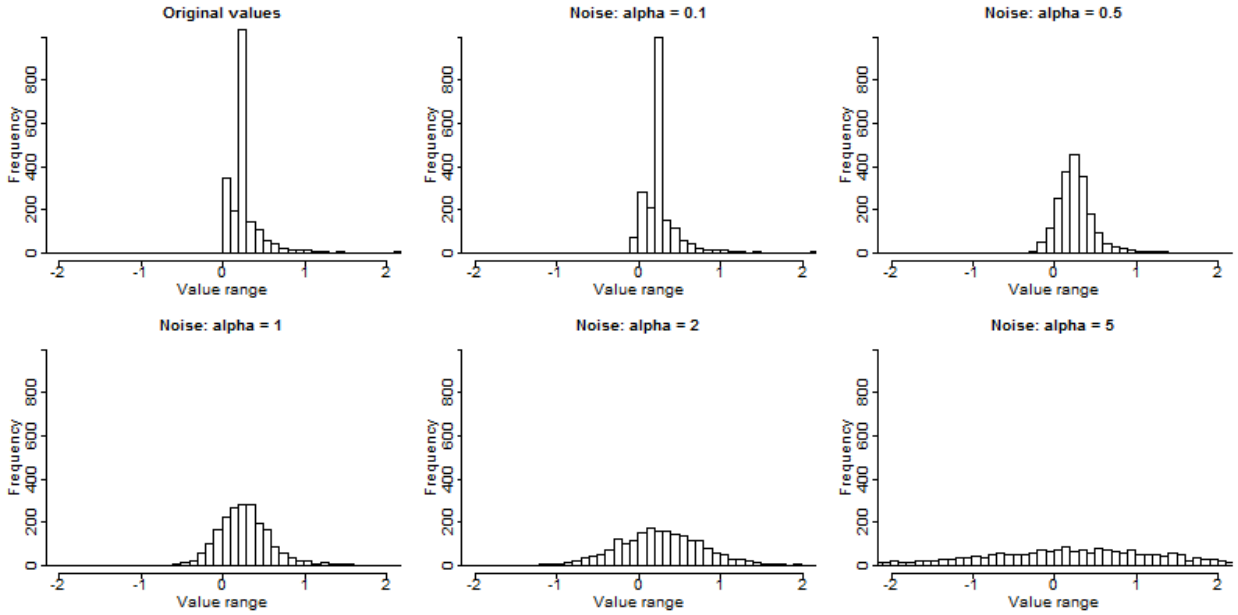

Fig. 6.7: Frequency distribution of a continuous variable before and after noise addition

Fig. 6.8 shows the value range of a variable before adding noise (no noise) and after adding several levels of noise (α from 0.1 to 1.5 with 0.1 increments). In the figure, the minimum value, the 20th, 30th, 40th percentiles, the median, the 60th, 70th, 80th and 90th percentiles and the maximum value are plotted. The median (50th percentile) is indicated with the red "+" symbol. From Fig. 6.7 and Fig. 6.8, it is apparent that the range of values expands after noise addition, and the median stays roughly at the same level, as does the mean by construction. The larger the magnitude of noise added, the wider the value range. In cases where the variable should stay in a certain value range (e.g., only positive values, between 0 and 100), this can be a disadvantage of noise addition. For instance, expenditure variables typically have non-negative values, but adding noise to these variables can generate negative values, which are difficult to interpret. One way to get around this problem is to set any negative values to zero. This truncation of values below a certain

[19] In this example and the following examples in this section, the *sdcMicro* object "sdcIntial" contains a dataset with 2,000 individuals and 39 variables. We selected five categorical quasi-identifiers and 12 continuous quasi-identifiers. These are the expenditure components "TFOODEXP", "TALCHEXP", "TCLTHEXP", "THOUSEXP", "TFURNEXP", "THLTHEXP", "TTRANSEXP", "TCOMMEXP", "TRECEXP", "TEDUEXP", "TRESTHOTEXP", "TMISCEXP".

threshold, however, will distort the distribution (mean and variance matrix) of the perturbed data. This means that the characteristics that were preserved by noise addition, such as the conservation of the mean and covariance matrix, are destroyed and the user, even with knowledge of the magnitude of the noise, can no longer use the data for consistent estimation.

Another way to avoid negative values is the application of multiplicative rather than additive noise. In that case, variables are multiplied by a random factor with expectation 1 and a positive variance. This will also lead to larger perturbations (in absolute value) of large initial values (outliers). If the variance of the noise added is small, there will be no or few negative factors and thus fewer sign changes than in case of additive noise masking. Multiplicative noise masking is not implemented in *sdcMicro*, but can be relatively easily implemented in base *R* by generating a vector of random numbers and multiplying the data with this vector. For more information on multiplicative noise masking and the properties of the data after masking, we refer to *KiWi03*.

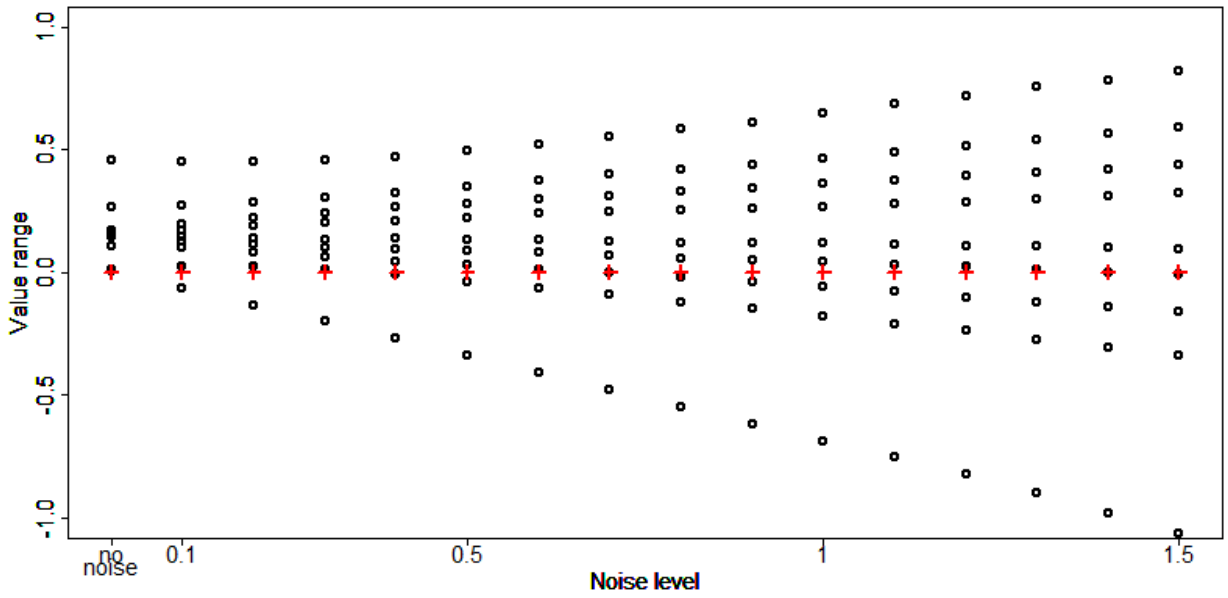

Fig. 6.8: Noise levels and the impact on the value range (percentiles)

If two or more variables are selected for noise addition, correlated noise addition is preferred to preserve the correlation structure in the data. In this case, the covariance matrix of noise Σ_ε is proportional to the covariance matrix of the original data Σ_X :

$$\Sigma_\varepsilon = \alpha\Sigma_X$$

In the addNoise() function of the *sdcMicro* package, correlated noise addition can be used by specifying the methods 'correlated' or 'correlated2'. The method "correlated" assumes that the variables are approximately normally distributed. The method 'correlated2' is a version of the method 'correlated', which is robust against the normality assumption. Listing 6.20 shows how to use the 'correlated2' method. The normality of variables can be investigated in *R*, with, for instance, a Jarque-Bera or Shapiro-Wilk test[20].

Listing 6.20: Correlated noise addition

```
sdcInitial <- addNoise(obj = sdcInitial,
                       variables = c('TOTFOOD', 'TOTHLTH', 'TOTALCH',
                                     'TOTCLTH', 'TOTHOUS', 'TOTFURN',
```

(continues on next page)

[20] The Shapiro-Wilk test is implemented in the function shapiro.test() from the package *stats* in *R*. The Jarque-Bera test has several implementations in *R*, for example, in the function jarque.bera.test() from the package *tseries*.

(continued from previous page)

```
                                        'TOTTRSP', 'TOTCMNQ', 'TOTRCRE',
                                        'TOTEDUC', 'TOTHOTL', 'TOTMISC'),
                          noise = 0.5, method = "correlated2")
```

In many cases, only the outliers have to be protected, or have to be protected more. The method 'outdect' adds noise only to the outliers, which is illustrated in Listing 6.21. The outliers are identified with univariate and robust multivariate procedures based on a robust Mahalanobis distance calculated by the MCD estimator (*TMKC14*). Nevertheless, noise addition is not the most suitable method for outlier protection.

Listing 6.21: Noise addition for outliers using the 'outdect' method

```
sdcInitial <- addNoise(obj = sdcInitial,
                          variables = c('TOTFOOD', 'TOTHLTH', 'TOTALCH',
                                        'TOTCLTH', 'TOTHOUS', 'TOTFURN',
                                        'TOTTRSP', 'TOTCMNQ', 'TOTRCRE',
                                        'TOTEDUC', 'TOTHOTL', 'TOTMISC'),
                          noise = 0.5, method = "outdect")
```

If noise addition is applied to variables that are a ratio of an aggregate, this structure can be destroyed by noise addition. Examples are income and expenditure data with many income and expenditure categories. The categories add up to total income or total expenditures. In the original data, the aggregates match with the sum of the components. After adding noise to their components (e.g., different expenditure categories), however, their new aggregates will not necessarily match the sum of the categories anymore. One way to keep this structure is to add noise only to the aggregates and release the components as ratio of the perturbed aggregates. Listing 6.22 illustrates this by adding noise to the total of expenditures. Subsequently, the ratios of the initial expenditure categories are used for each individual to reconstruct the perturbed values for each expenditure category.

Listing 6.22: Noise addition to aggregates and their components

```
# Add noise to totals (income / expenditures)
sdcInital <- addNoise(noise = 0.5, obj = sdcInitial, variables=c("EXP", "INC"),
                      method="additive")

# Multiply anonymized totals with ratios to obtain anonymized components
compExp <-  c("TOTFOOD",  "TOTALCH",  "TOTCLTH",  "TOTHOUS",  "TOTFURN",
              "TOTHLTH",  "TOTTRSP",  "TOTCMNQ", "TOTRCRE",  "TOTEDUC",
              "TOTHOTL",  "TOTMISC")

sdcInital@manipNumVars[,compExp] <- sdcInital@manipNumVars[,"HHEXP_N"] *
                            sdcInital@origData[,compExp]/ sdcInital@origData[,"HHEXP_
↪N"]

# Recalculate risks after manually changing values in sdcMicro object
sdcInitial <- calcRisks(sdcInital)
```

Recommended Reading Material on Noise Addition

Brand, Ruth. 2002. "Microdata Protection through Noise Addition." In *Inference Control in Statistical Databases - From Theory to Practice*, edited byJosep Domingo-Ferrer. Lecture Notes in Computer Science Series 2316, 97-116. Berlin Heidelberg: Springer. http://link.springer.com/chapter/10.1007%2F3-540-47804-3_8

Kim, Jay J, and William W Winkler. 2003. "Multiplicative Noise for Masking Continuous Data." *Research Report Series* (Statistical Research Division. US Bureau of the Census). https://www.census.gov/srd/papers/pdf/rrs2003-01.pdf

Torra, Vicenç, and Isaac Cano. 2011. "Edit Constraints on Microaggregation and Additive Noise." In *Privacy and Se-*

curity Issues in Data Mining and Machine Learning, edited by C. Dimitrakakis, A. Gkoulalas-Divanis, A. Mitrokotsa, V. S. Verykios, Y. Saygin. Lecture Notes in Computer Science Volume 6549, 1-14. Berlin Heidelberg: Springer. http://link.springer.com/book/10.1007/978-3-642-19896-0

Mivule, K. 2013. "Utilizing Noise Addition for Data Privacy, An Overview." *Proceedings of the International Conference on Information and Knowledge Engineering (IKE 2012)*, (pp.65-71).Las Vegas, USA. http://arxiv.org/ftp/arxiv/papers/1309/1309.3958.pdf

6.3.4 Rank swapping

Data swapping is based on interchanging values of a certain variable across records. Rank swapping is one type of data swapping, which is defined for ordinal and continuous variables. For rank swapping, the values of the variable are first ordered. The possible number of values for a variable to swap with is constrained by the values in a neighborhood around the original value in the ordered values of the dataset. The size of this neighborhood can be specified, e.g., as a percentage of the total number of observations. This also means that a value can be swapped with the same or very similar values. This is especially the case if the neighborhood is small or there are only a few different values in the variable (ordinal variable). An example is the variables "education" with only few categories: ('none', 'primary', 'secondary', 'tertiary'). In these cases, rank swapping is not a suitable method.

If rank swapping is applied to several variables simultaneously, the correlation structure between the variables is preserved. Therefore, it is important to check whether the correlation structure in the data is plausible. Rank swapping is implemented in the function rankSwap() in *sdcMicro*. The variables, which have to be swapped, should be specified in the argument 'variables'. By default, values below the 5th percentile and above the 95th percentile are top and bottom coded and replaced by their average value (see the Section Top and bottom coding). By specifying the options 'TopPercent' and 'BottomPercent' we can choose these percentiles. The argument 'P' defines the size of the neighborhood as percentage of the sample size. If the value 'p' is 0.05, the neighborhood will be of size 0.05 * n, where n is the sample size. Since rank swapping is a probabilistic method, i.e., the swapping depends on a random number generating mechanism, specifying a seed for the random number generator before using rank swapping is recommended to guarantee reproducibility of results. The seed can also be specified as a function argument in the function rankSwap(). Listing 6.23 shows how to apply rank swapping with *sdcMicro*. If the variables contain missing values (NA in *R*), the function rankSwap() will automatically recode those to the value specified in the 'missing' argument. This value should not be in the value range of any of the variables. After using the function rankSwap(), these values should be recoded NA. This is shown in the Listing 6.23.

Listing 6.23: Rank swapping using *sdcMicro*

```
# Check correlation structure between the variables
cor(file$TOTHOUS, file$TOTFOOD)
## [1] 0.3811335

# Set seed for random number generator
set.seed(12345)

# Apply rank swapping
rankSwap(sdcInitial, variables = c("TOTHOUS", "TOTFOOD"), missing = NA)
```

Rank swapping has been found to yield good results with respect to the trade-off between information loss and data protection (*DoTo01a*). Rank swapping is not useful for variables with few different values or many missing values, since the swapping in that case will not result in altered values. Also, if the intruder knows to whom the highest or lowest value of a specific variable belongs (e.g., income), the level of this variable will be disclosed after rank swapping, because the values themselves are not altered and the original values are all disclosed. This can be solved by top and bottom coding the lowest and/or highest values.

Recommended Reading Material on Rank Swapping

Dalenius T. and Reiss S.P. 1978. Data-swapping: a technique for disclosure control (extended abstract). In Proc. ASA Section on Survey Research Methods. American Statistical Association, Washington DC, 191–194.

Domingo-Ferrer J. and Torra V. 2001. "A Quantitative Comparison of Disclosure Control Methods for Microdata." In *Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies*, edited by P. Doyle, J.I. Lane, J.J.M. Theeuwes, and L. Zayatz, 111–134. Amsterdam, North-Holland.

Hundepool A., Van de Wetering A., Ramaswamy R., Franconi F., Polettini S., Capobianchi A., De Wolf P.-P., Domingo-Ferrer J., Torra V., Brand R. and Giessing S. 2007. μ-Argus User's Manual version 4.1.

6.3.5 Shuffling

Shuffling as introduced by *MuSa06* is similar to swapping, but uses an underlying regression model for the variables to determine which variables are swapped. Shuffling can be used for continuous variables and is a deterministic method. Shuffling maintains the marginal distributions in the shuffled data. Shuffling, however, requires a complete ranking of the data, which can be computationally very intensive for large datasets with several variables.

The method is explained in detail in *MuSa06*. The idea is to rank the individuals based on their original variables. Then fit a regression model with the variables to be protected as regressands and a set of variables that predict this variable well (i.e., are correlated with) as regressors. This regression model is used to generate n synthetic (predicted) values for each variable that has to be protected. These generated values are also ranked and each original value is replaced with another original value with the rank that corresponds to the rank of the generated value. This means that all original values will be in the data. Table 6.14 presents a simplified example of the shuffling method. The regressands are not specified in this example.

Table 6.14: Simplified example of the shuffling method

ID	Income (orig)	Rank (orig)	Income (pred)	Rank (pred)	Shuffled values
1	2,300	2	2,466.56	4	2,345
2	2,434	6	2,583.58	7	2,543
3	2,123	1	2,594.17	8	2,643
4	2,312	3	2,530.97	6	2,434
5	6,045	10	5,964.04	10	6,045
6	2,345	4	2,513.45	5	2,365
7	2,543	7	2,116.16	1	2,123
8	2,854	9	2,624.32	9	2,854
9	2,365	5	2,203.45	2	2,300
10	2,643	8	2,358.29	3	2,312

The method 'ds' (the default method of data shuffling in *sdcMicro*) is recommended for use (*TeMK14*)[21]. A regression function with regressors for the variables to be protected must be specified in the argument 'form'. At least two regressands should be specified and the regressors should have predictive power for the variables to be predicted. This can be checked with goodness-of-fit measures such as the R^2 of the regression. The R^2 captures only linear relations, but these are also the only relations that are captured by the linear regression model used for shuffling. Following is an example for shuffling expenditure variables, which are predicted by total household expenditures and household size.

Listing 6.24: Shuffling using a specified regression equation

```
# Evaluate R-squared (goodness-of-fit) of the regression model
summary(lm(file, form = TOTFOOD  + TOTALCH + TOTCLTH + TOTHOUS +
                        TOTFURN + TOTHLTH  + TOTTRSP + TOTCMNQ +
```

(continues on next page)

[21] In *sdcMicro*, there are several other methods for shuffling implemented, including 'ds', 'mvn' and 'mlm'. See the Help option for the shuffle function in *sdcMicro* for details on methods 'ds', 'mvm' and 'mlm'.

(continued from previous page)

```
                          TOTRCRE + TOTEDUC + TOTHOTL + TOTMISC ~ EXP + HHSIZE))

# Shuffling using the specified regression equation
sdcInitial <- shuffle(sdcInitial, method='ds',
                      form = TOTFOOD  + TOTALCH + TOTCLTH + TOTHOUS +
                             TOTFURN + TOTHLTH  + TOTTRSP + TOTCMNQ +
                             TOTRCRE + TOTEDUC + TOTHOTL + TOTMISC ~ EXP + HHSIZE)
```

Recommended Reading Material on Shuffling

K. Muralidhar and R. Sarathy. 2006."Data shuffling - A new masking approach for numerical data," *Management Science*, 52, 658-670.

6.3.6 Comparison of PRAM, rank swapping and shuffling

PRAM, rank swapping and shuffling are all perturbative methods, i.e., they change the values for individual records and are mainly used for continuous variables. After rank swapping and shuffling, the original values are all contained in the treated dataset but might be assigned to other records. This implies that univariate tabulations are not changed. This also holds in expectation for PRAM, if a transition matrix is chosen that has the invariant property.

Choosing a method is based on the structure to be preserved in the data. In cases where the regression model fits the data well, data shuffling would work very well, as there should be sufficient (continuous) regressors available. Rank swapping works well if there are sufficient categories in the variables. PRAM is preferred if the perturbation method should be applied to only one or few variables; the advantage is the possibility of specifying restrictions on the transition matrix and applying PRAM only within strata, which can be user defined.

6.4 Anonymization of geospatial variables

Recently, geospatial data has become increasingly popular with researchers and wide-spread. Georeferenced data identifies the geographical location for each record with the help of a Geographical Information System (GIS), that uses for instance GPS (Global Positioning System) coordinates or address data. The advantages of geospatial data are manifold: 1) researchers can create their own geographical areas, such as the service area of a hospital; 2) it enables researchers to measure the proximity to facilities, such as schools; 3) researchers can use the data to extract geographical patterns; and 4) it enables linking of data from different sources (see e.g., *BCRZ13*). However, geospatial data, due to the precise reference to a location, also pose a challenge to the privacy of the respondents.

One way to anonymize georeferenced data is removing the GIS variables and instead leaving in or creating other geographical variables, such as province, region. However, this approach also removes the benefits of geospatial data. Another option is the geographical displacement of areas and/or records. *BCRZ13* describe a geographical displacement procedure for a health dataset. This paper also includes the code in Python. *HuDr15* propose three different strategies for generating synthetic geocodes.

Recommended Reading Material on Anonymization of Geospatial Data

C.R. Burgert, J. Colston, T. Roy and B. Zachary. 2013. "DHS Spatial Analysis Report No. 7 - Geographic Displacement Procedure and Georeferenced Data Release Policy for the Demographic and Health Surveys" (USAID). http://dhsprogram.com/pubs/pdf/SAR7/SAR7.pdf

J. Hu and J. Drechsler. 2015. "Generating synthetic geocoding information for public release." http://www.iab.de/389/section.aspx/Publikation/k150601301

6.5 Anonymization of the quasi-identifier household size

The size of a household is an important identifier, especially for large households.[22] Suppression of the actual size variable, if available (e.g., number of household members), however, does not suffice to remove this information from the dataset, as a simple count of the household members for a particular household will allow this variable to be reconstructed as long as a household ID is in the data. In any case, households of a very large size or with a unique or special key (i.e., combination of values of quasi-identifiers) should be checked manually. One way to treat them is to remove these households from the dataset before release. Alternatively, the households can be split, but care should be taken to suppress or change values for these households to prevent an intruder from immediately understanding that these households have been split and reconstructing them by combining the two households with the same values.

6.6 Special case: census data

Census microdata are a special case because the user (and intruder) knows that all respondents are included in the dataset. Therefore, risk measures that use the sample weights and are based on uncertainty of the correctness of a match are no longer applicable. If an intruder has identified a sample unique and successfully matched, there is no doubt whether the match is correct, as it would be in the case of a sample. One approach to release census microdata is to release a stratified sample of the sample (1 – 5% of the total census).

Note: After sampling, the anonymization process has to be followed; sampling alone is not sufficient to guarantee confidentiality.

Several statistical offices release microdata based on census data. A few examples are:

- **The British Office for National Statistics (ONS)** released several files based on the 2011 census: 1. A microdata teaching file for educational purposes. This file is a 1% sample of the total census with a limited set of variables. 2. Two scientific use files with 5% samples are available for registered researchers who accept the terms and conditions of their use. 3. Two 10% samples are available in controlled research data centers for approved researchers and research goals. All these files have been anonymized prior to release.[23]
- **The U.S. Census Bureau** released two samples of the 2000 census: a 5% sample on the national level and a 1% sample on the state level. The national level file is more detailed, but the most detailed geographical area has at least 400,000 people. This, however, allows representation of all states from the dataset. The state-level file has less detailed variables but a more detailed geographical structure, which allows representation of cities and larger counties from the dataset (the minimum size of a geographical area is 100,000). Both files have been anonymized by using data swapping, top coding, perturbation and reducing detail by recoding.[24]

References

[22] Even if the dataset does not contain an explicit variable with household size, this information can be easily extracted from the data and should be taken into account. The Section Household structure shows how to create a variable "household size" based on the household IDs.

[23] More information on census microdata at ONS is available on their website: http://www.ons.gov.uk/ons/guide-method/census/2011/census-data/census-microdata/index.html

[24] More information on the anonymization of these files is available on the website of the U.S. Census Bureau: https://www.census.gov/population/www/cen2000/pums/index.html

CHAPTER

SEVEN

MEASURING UTILITY AND INFORMATION LOSS

SDC is a trade-off between risk of disclosure and loss of data utility and seeks to minimize the latter, while reducing the risk of disclosure to an acceptable level. Data utility in this context means the usefulness of the anonymized data for statistical analyses by end users as well as the validity of these analyses when performed on the anonymized data. Disclosure risk and its measurement are defined in the Section Measure Risk of this guide. In order to make a trade-off between minimizing disclosure risk and maximizing utility of data for end users, it is necessary to measure the utility of the data after anonymization and compare it with the utility of the original data. This section describes measures that can be used to compare the data utility before and after anonymization, or alternatively quantify the information loss. Information loss is the inverse of data utility: the larger the data utility after anonymization, the smaller the information loss.

Note: If the microdata to be anonymized is based on a sample, the data will incur a sampling error. Also other errors may be present in the data, such as nonresponse error.

Note: The methods discussed here only measure the information loss caused by the anonymization process relative to the original sample data and do not attempt to measure the error caused by other sources.

Ideally, the information loss is evaluated with respect to the needs and uses of the end users of the microdata. However, different end users of anonymized data may have very diverse uses for the released data and it might not be possible to collect an exhaustive list of these uses. Even if many uses can be identified, the characteristics in the data needed for these uses can be contradictory (e.g., one user needs a detailed geographical level whereas another is interested in a detailed age structure and does not need a detailed geographical structure). Nevertheless, as pointed out earlier, only one anonymized dataset can be released for each dataset and every type of release to avoid unintended disclosure. Releasing multiple anonymized datasets for different purposes may lead to unintended disclosure.[1] Therefore, it is not possible to anonymize and release a file tailored to each user's needs.

Since collecting and taking into account all data uses is often impossible, we also look at general (or generic) information loss measures besides user- and data-specific information loss measures. These measures do not take into account the specific data use, but can be used as guiding measures for information loss and evaluating whether a dataset is still analytically valid after anonymization. The main idea for such measures is to compare records between the original and treated datasets and compare statistics computed from both datasets (*HDFG12*). Examples of such measures are the number of suppressions, number of changed values, changes in contingency tables and changes in mean and covariance matrices.

Many of the SDC methods discussed earlier are parametric, in the sense that their outcome depends on parameters chosen by the user. Examples are the cluster size for microaggregation (see the Section Microaggregation) or the importance vector in local suppression (see the Section Local suppression). Data utility and information loss measures

[1] It is possible to release data files for different groups of users, e.g., PUF and SUF. All information in the less detailed file, however, must also be included in the more detailed file to prevent unintended disclosure. Datasets released in data enclaves can be customized for the user, since the risk that they will be combined with other version is zero.

are useful for choosing these parameters by comparing the impact of different parameters on the information loss. Fig. 7.1 illustrates this by showing the trade-off between the disclosure risk and data utility of a hypothetical dataset. The triangle represents the original data with full utility and a certain level of disclosure risk, which is too high for disclosure. The square represents no release of microdata. Although there is no risk of disclosure, there is also no utility from the data for users since no data is released. The points in between represent the result of applying different SDC methods with different parameter specifications. We would select the SDC method corresponding to the point, which maximizes the utility, while keeping disclosure risk at an acceptable level.

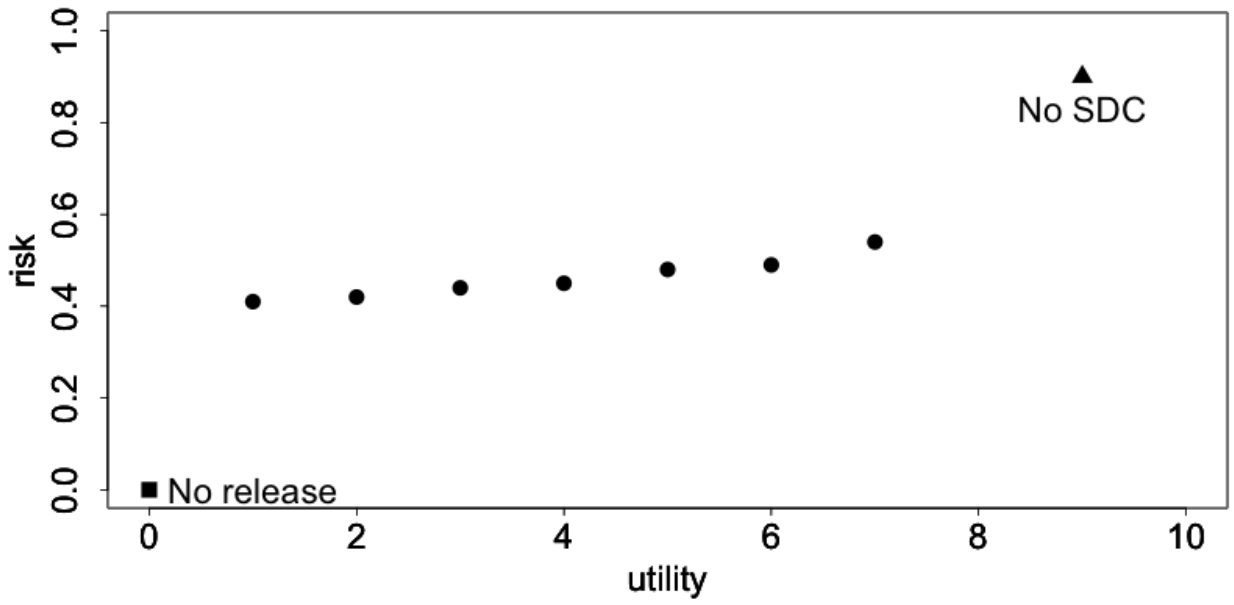

Fig. 7.1: The trade-off between risk and utility in a hypothetical dataset

In the following sections, we first propose general utility measures independent of data use, and later present an example of a specific measure useful to measure information loss with respect to specific data uses. Finally, we show how to visualize changes in the data caused by anonymization and discuss the selection of utility measures for a particular dataset.

7.1 General utility measures for continuous and categorical variables

General or generic measures of information loss can be divided into those comparing the actual values of the raw and anonymized data, and those comparing statistics from both datasets. All measures are a posteriori, since they measure utility after anonymization and require both the data before and after the anonymization process. General utility measures are different for categorical and continuous variables.

7.1.1 General utility measures for categorical variables

Number of missing values

An informative measure is to compare the number of missing values in the data. Missing values are often introduced after suppression and more suppressions indicate a higher degree of information loss. After using the local suppression function on an *sdcMicro* object, the number of suppressions for each categorical key variable can be retrieved with

the print() function, which is illustrated in Listing 7.1[2]. The argument 'ls' in the print() function stands for local suppression. The output shows both the absolute and relative number of suppressions.

Listing 7.1: Using the print() function to retrieve the total number of suppressions for each categorical key variable

```
sdcInitial <- localSuppression(sdcInitial, k = 5, importance = NULL)

print(sdcInitial, 'ls')

## Local Suppression:
##   KeyVar | Suppressions (#) | Suppressions (%)
##   URBRUR |                 0 |            0.000
##   REGION |                81 |            4.050
##    RELIG |                 0 |            0.000
##  MARITAL |                 0 |            0.000
## ----------------------------------------------------------------------------
```

More generally, it is possible to count and compare the number of missing values in the original data and the treated data. This can be useful to see the proportional increase in the number of missing values. Missing values can also have other sources, such as nonresponse. Listing 7.2 shows how to display the number of missing values for each of the categorical key variables in an *sdcMicro* object. Here it is assumed that all missing values are coded 'NA'. If the missing values are not coded 'NA', but instead another value, it is possible to use the alternative missing values code. The results agree with the number of missing values introduced by local suppression in the previous example, but also shows that the variable "RELIG" has 1,000 missing values in the original data.

Listing 7.2: Displaying the number of missing values for each categorical key variable in an *sdcMicro* object

```
# Store the names of all categorical key variables in a vector
namesKeyVars <- names(sdcInitial@manipKeyVars)

# Matrix to store the number of missing values (NA) before and after anonymization
NAcount <- matrix(NA, nrow = 2, ncol = length(namesKeyVars))
colnames(NAcount) <- c(paste0('NA', namesKeyVars)) # column names
rownames(NAcount) <- c('initial', 'treated') # row names

# NA count in all key variables (NOTE: only those coded NA are counted)
for(i in 1:length(namesKeyVars)) {
  NAcount[1, i] <- sum(is.na(sdcInitial@origData[,namesKeyVars[i]]))
  NAcount[2, i] <- sum(is.na(sdcInitial@manipKeyVars[,i]))
}

# Show results
NAcount
## NAURBRUR NAREGION NARELIG NAMARITAL
## initial 0 0 1000 51
## treated 0 81 1000 51
```

Number of records changed

Another useful statistic is the number of records changed per variable. These can be counted in a similar way as the missing values and include suppressions (i.e., changes to missing/'NA' in *R*). The number of records changed gives a

[2] Here the *sdcMicro* object "sdcIntial" contains a dataset with 2,500 individuals and 103 variables. We selected three categorical quasi-identifiers: "URBRUR", "REGION", "RELIG" and "MARITAL" and several continuous quasi-identifiers relating to income and expenditure. To illustrate the utility loss, we also applied several SDC methods to this *sdcMicro* object, such as local suppression, PRAM and additive noise addition.

good indication of the impact of the anonymization methods on the data. Listing 7.3 illustrates how to compute the number of records changed for the PRAMmed variables.

Listing 7.3: Computing number of records changed per variable

```
# Store the names of all pram variables in a vector
namesPramVars <- names(sdcInitial@manipPramVars)

# Dataframe to save the number of records changed
recChanged <- rep(0, length(namesPramVars))
names(recChanged) <- c(paste0('RC', namesPramVars))

# Count number of records changed
for(j in 1:length(namesPramVars)) # for all key variables
{
  comp <- sdcInitial@origData[namesPramVars[j]] !=
                              sdcInitial@manipPramVars[namesPramVars[j]]
  temp1 <- sum(comp, na.rm = TRUE) # all changed variables without NAs
  temp2 <- sum(is.na(comp))        # if NA, changed, unless NA initially
  temp3 <- sum(is.na(sdcInitial@origData[namesPramVars[j]])
               + is.na(sdcInitial@manipPramVars[namesPramVars[j]])==2)
  # both NA, no change, but counted in temp2
  recChanged[j] <- temp1 + temp2 - temp3
}

# Show results
recChanged
##  RCWATER   RCROOF RCTOILET
##      125       86      180
```

Comparing contingency tables

A useful way to measure information loss in categorical variables is to compare univariate tabulations and, more interestingly, contingency tables (also cross tabulations or two-way tables) between pairs of variables. To maintain the analytical validity of a dataset, the contingency tables should stay approximately the same. The function table() produces contingency tables of one or more variables. Listing 7.4 creates a contingency table of the variables "REGION" and "URBRUR". We observe small differences between the tables before and after anonymization.

Listing 7.4: Comparing contingency tables of categorical variables

```
 # Contingency table (cross tabulation) of the variables region and urban/rural
 table(sdcInitial@origData[, c('REGION', 'URBRUR')]) # before anonymization
 ##       URBRUR
 ## REGION   1   2
 ##      1 235  89
 ##      2 261  73
 ##      3 295  76
 ##      4 304  71
 ##      5 121 139
 ##      6 100 236

 table(sdcInitial@manipKeyVars[, c('REGION', 'URBRUR')]) # after anonymization
 ##       URBRUR
 ## REGION   1   2
 ##      1 235  89
 ##      2 261  73
```

(continues on next page)

(continued from previous page)

```
##       3 295  76
##       4 304  71
##       5 105 130
##       6  79 201
```

DoTo01b propose a Contingency Table-Based Information Loss (CTBIL) measure, which quantifies the distance between the contingency tables in the original and treated data. Alternatively, visualizations of the contingency table with mosaic plots can be used to compare the impact of anonymization methods on the tabulations and contingency tables (see the Section *Mosaic plots*).

7.1.2 General utility measures for continuous variables

Statistics: mean, covariance, correlation

The statistics characterizing the dataset should not change after the anonymization. Examples of such statistics are the mean, variance, and covariance and correlation structure of the most important variables in the dataset. Other statistics characterizing the data include the principal components and the loadings. *DoTo01b* give an overview of statistics that can be considered. In order to evaluate the information loss caused by the anonymization, one should compare the appropriate statistics for continuous variables computed from the data before and after anonymization. There are several ways to evaluate the loss of utility with respect to the changes in these statistics, for instance, by comparing means and (co-)variances in the data or comparing the (multivariate) distributions of the data. Especially changes in the correlations gives valuable information on the validity of the data for regressions. Functions from the *R* base package or any other statistical package can be used to do this. Following are a few examples in *R*.

To compute the mean of each numerical variable we use the function colMeans(). To ignore missing values, it is necessary to use the option na.rm = TRUE. "numVars" is a vector with the names of the numerical variables. Listing 7.5 shows how to compute the means for all numeric variables. The untreated data is extracted from the 'origData' slot of the *sdcMicro* object and the anonymized data from the 'manipNumVars' slot, which contains the manipulated numeric variables. We observe small changes in each of the three variables.

Listing 7.5: Comparing the means of continuous variables

```
# untreated data
colMeans(sdcInitial@origData[, numVars], na.rm = TRUE)
##        INC    INCRMT   INCWAGE
##  479.7710  961.0295 1158.1330

# anonymized data
colMeans(sdcInitial@manipNumVars[, numVars], na.rm = TRUE)
##        INC    INCRMT   INCWAGE
##  489.6030  993.8512 1168.7561
```

In the same way, one can compute the covariance and correlation matrices of the numerical variables in the *sdcMicro* object from the untreated and anonymized data. This is shown in Listing 7.6. We observe that the variance of each variable (the diagonal elements in the covariance matrix) have increased by the anonymization. These functions also allow computing confidence intervals in the case of samples. The means and covariances of subsets in the data also should not differ. An example is the mean of income by gender, by age group or by region. These characteristics of the data are important for analysis.

Listing 7.6: Comparing covariance structure and correlation matrices of numeric variables

```
# untreated data
cov(sdcInitial@origData[, numVars])
##                INC    INCRMT  INCWAGE
## INC     1645926.1  586975.6  2378901
## INCRMT   586975.6 6984502.3  1664257
## INCWAGE 2378900.7 1664257.4 16169878

cor(sdcInitial@origData[, numVars])

##                INC    INCRMT   INCWAGE
## INC     1.0000000 0.1731200 0.4611241
## INCRMT  0.1731200 1.0000000 0.1566028
## INCWAGE 0.4611241 0.1566028 1.0000000

# anonymized data
cov(sdcInitial@manipNumVars[, numVars])
##                INC    INCRMT  INCWAGE
## INC     2063013.1  649937.5  2382447
## INCRMT   649937.5 8566169.1  1778985
## INCWAGE 2382447.4 1778985.1 19925870

cor(sdcInitial@manipNumVars[, numVars])
##                INC    INCRMT   INCWAGE
## INC     1.0000000 0.1546063 0.3715897
## INCRMT  0.1546063 1.0000000 0.1361665
## INCWAGE 0.3715897 0.1361665 1.0000000
```

DoTo01b propose several measures for the discrepancy between the covariance and correlation matrices. These measures are based on the mean squared error, the mean absolute error or the mean variation of the individual cells. We refer to *DoTo01b* for a complete overview of these measures.

IL1s information loss measure

Alternatively, we can also compare the actual data and quantify the distance between the original dataset X and the treated dataset Z. Here X and Z contain only continuous variables. *YaWC02* introduce the distance measure IL1s, which is the sum of the absolute distances between the corresponding observations in the raw and anonymized datasets, which are standardized by the standard deviation of the variables in the original data. For the continuous variables in the dataset, the IL1s measure is defined as

$$IL1s = \frac{1}{\text{pn}} \sum_{j=1}^{p} \sum_{i=1}^{n} \frac{|x_{\text{ij}} - z_{\text{ij}}|}{\sqrt{2} S_j},$$

where p is the number of continuous variables; n is the number of records in the dataset; x_{ij} and z_{ij}, respectively, are the values before and after anonymization for variable j and individual i; and S_j is the standard deviation of variable j in the original data (*YaWC02*).

When using *sdcMicro*, the IL1s data utility measure can be computed for all numerical quasi-identifiers with the function dUtility(), which is illustrated in Listing 7.7. If required, the measure can also be computed on subsets of the complete set of numerical quasi-identifiers. The function is called dUtility(), but returns a measure of information loss. The result is saved in the utility slot of the *sdcMicro* object. Listing 7.7 also illustrates how to call the result.

Listing 7.7: Using dUtility() to compute IL1s data utility measure in *sdcMicro*

```
# Evaluating IL1s measure for all variables in the sdcMicro object sdcInitial
sdcInitial <- dUtility(sdcInitial)

# Calling the result of IL1s
sdcInitial@utility$il1
## [1] 0.2203791

# IL1s for a subset of the numerical quasi-identifiers
subset <- c('INCRMT', 'INCWAGE', 'INCFARMBSN')
dUtility(obj = sdcInitial@origData[,subset], xm = sdcInitial@manipNumVars[,subset],
method = 'IL1')
## [1] 0.5641103
```

The measure is useful for comparing different methods. The smaller the value of the measure, the closer the values are to the original values and the higher the utility.

Note: This measure is related to risk measures based on distance and intervals (see the Section Risk measures for continuous variables).

The greater the distance between the original and anonymized values, the lower the data utility. Greater distance, however, also reduces the risk of re-identification.

Eigenvalues

Another way to evaluate the information loss is to compare the robust eigenvalues of the data before and after anonymization. Listing 7.8 illustrates how to use this approach with *sdcMicro*. Here "contVars" is a vector with the names of the continuous variables in which we are interested. "obj" is the argument that specifies the untreated data and "xm" is the argument that specifies the anonymized data. The function's output is the difference in eigenvalues. Therefore, the minimum value is 0. Again, the main use is to compare different methods. The greater the value, the greater the changes in the data and the information loss.

Listing 7.8: Using dUtility() to compute eigenvalues in *sdcMicro*

```
# Comparison of eigenvalues of continuous variables
dUtility(obj = sdcInitial@origData[,contVars],
         xm = sdcInitial@manipNumVars[,contVars], method = 'eigen')
## [1] 2.482948

# Comparison of robust eigenvalues of continuous variables*
dUtility(obj = sdcInitial@origData[,contVars],
         xm = sdcInitial@manipNumVars[,contVars], method = 'robeigen')
## [1] -4.297621e+14
```

7.2 Utility measures based on the end user's needs

Not all needs and uses of a certain dataset can be inventoried. Nevertheless, some types of data have similar uses or important characteristics, which can be evaluated before and after anonymization. Examples of such "benchmarking indicators" (*TMKC14*) are different for each dataset. Examples include poverty measures for income datasets and

school attendance ratios. Often ideas for selecting such indicators come from the reports data users publish based on previously released microdata.

The approach is to compare the indicators calculated on the untreated data and the data after anonymization with different methods. If the differences between the indicators are not too large, the anonymized dataset can be released for use by researchers. It should be taken into account that indicators calculated on samples are estimates with a certain variance and confidence interval. Therefore, for sample data, it is more informative to compare the overlap of confidence intervals and/or to evaluate whether the point estimate calculated after anonymization is contained within the confidence interval of the original estimate. Examples of benchmark indicators and their confidence intervals and how to compute these in *R* are included in the case studies in these guidelines. Here we give the example of the GINI coefficient.

The GINI coefficient is a measure of statistical dispersion, which is often used to measure inequality in income. A way to measure the information loss in income data is to compare the income distribution, which can be easily done by comparing the GINI coefficients. Several *R* packages have functions to compute the GINI coefficient. We chose the *laeken* package, which computes the GINI coefficient as the area between the 45-degree line and the Lorenz curve. To use the gini() function, we first have to install and load the *laeken* library. To calculate the GINI coefficient for the variable income, we use the sample weights in the data. This is shown in Listing 7.9. The GINI coefficient of sample data is a random variable. Therefore, it is useful to construct a confidence interval around the coefficient to evaluate the significance of any change in the coefficient after anonymization. The gini() function computes a 1-alpha confidence interval for the GINI coefficient by using bootstrap.

Listing 7.9: Computing the GINI coefficient from the income variable to determine income inequality

```
# Gini coefficient before anonymization
gini(inc = sdcInitial@origData[selInc,'INC'],
     weights =  curW[selInc], na.rm = TRUE)$value # before
## [1] 34.05928

# Gini coefficient after anonymization
gini(inc = sdcInitial@manipNumVars[selInc,'INC'],
     weights = curW[selInc], na.rm = TRUE)$value # after
## [1] 67.13218
```

7.3 Regression

Besides comparing covariance and correlation matrices, regressions are a useful tool to evaluate whether the structure in the data is maintained after anonymization. By comparing regressions parameters, it is also possible to compare relations between non-continuous variables (e.g., by introducing dummy variables or regression with ordinal variables). If it is known for what purpose and in what field the data is used, common regressions can be used to compare the change in coefficients and confidence intervals.

An example of using regression to evaluate the data utility in income data is the Mincer equation. The Mincer equation explains earnings as a function of education and experience while controlling for other variables. The Mincer equation is often used to evaluate the gender pay gap and gender wage inequality by including a gender dummy. Here we show how to evaluate the impact of anonymization methods on the gender coefficient. We regress the log income on a constant, a gender dummy, years of education, years of experience, years of experience squared and other factors influencing wage.

$$\ln(\text{wage}) = \beta_0 + \beta_1 gender + \beta_2 education + \beta_3 experience + \beta_3 \text{experience}^2 + \beta X$$

The parameter of interest here is β_1, the effect of gender on the log wage. X is a matrix with several other factors influencing wage and β the coefficients of these factors. Listing 7.10 illustrates how to run a Mincer regression in

R using the function lm() and evaluate the coefficients and confidence intervals around the coefficients. We run the regression as specified for paid employees with a positive wage in the age groups 15 – 65 years.

Listing 7.10: Estimating the Mincer equation (regression) to evaluate data utility before and after anonymization

```
# Mincer equation variables before anonymization
Mlwage    <- log(sdcMincer@origData$wage) # log wage
# TRUE if 'paid employee', else FALSE or NA
Mempstat  <- sdcMincer@origData$empstat=='Paid employee'
Mage      <- sdcMincer@origData$age    # age in years
Meducy    <- sdcMincer@origData$educy  # education in years
Mexp      <- sdcMincer@origData$exp    # experience in years
Mexp2     <- Mexp^2                    # squared experience
Mgender   <- sdcMincer@origData$gender # gender dummy
Mwgt      <- sdcMincer@origData$wgt    # weight variable for regression
MfileB    <- as.data.frame(cbind(Mlwage, Mempstat, Mage, Meducy, Mexp, Mexp2,
                                 Mgender, Mwgt))
# Mincer equation variables after anonymization
Mlwage    <- log(sdcMincer@manipNumVars$wage) # log wage
Mempstat  <- sdcMincer@manipKeyVars$empstat=='Paid employee'
# TRUE if 'paid employee', else FALSE or NA
Mage      <- sdcMincer@manipKeyVars$age    # age in years
Meducy    <- sdcMincer@manipKeyVars$educy  # education in years
Mexp      <- sdcMincer@manipKeyVars$exp    # experience in years
Mexp2     <- Mexp^2                        # squared experience
Mgender   <- sdcMincer@manipKeyVars$gender # gender dummy
Mwgt      <- sdcMincer@origData$wgt        # weight variable for regression
MfileA    <- as.data.frame(cbind(Mlwage, Mempstat, Mage, Meducy, Mexp, Mexp2,
                                 Mgender, Mwgt))

# Specify regression formula
Mformula <- 'Mlwage ~ Meducy + Mexp + Mexp2 + Mgender'

# Regression Mincer equation
mincer1565B <- lm(Mformula, data = subset(MfileB,
MfileB$Mage >= 15 & MfileB$Mage <= 65 & MfileB$Mempstat==TRUE &
MfileB$Mlwage != -Inf), na.action = na.exclude, weights = Mwgt) # before
mincer1565A <- lm(Mformula,
                  data = subset(MfileA,
                                                MfileA$Mage >= 15 & MfileA$Mage <= 65 &
                                                MfileA$Mempstat==TRUE &
                                MfileA$Mlwage != -Inf),
                  na.action = na.exclude, weights = Mwgt) # after

# The objects mincer1565B and mincer1565A contain the results of the
regressions before and after anonymization
mincer1565B$coefficients # before
##   (Intercept)        Meducy          Mexp         Mexp2       Mgender
##  3.9532064886  0.0212367075  0.0255962570 -0.0005682651 -0.4931289413

mincer1565A$coefficients # after
##   (Intercept)        Meducy          Mexp         Mexp2       Mgender
##  4.0526250282  0.0141090329  0.0326711056 -0.0007605492 -0.5393641862

# Compute the 95 percent confidence interval
confint(obj = mincer1565B, level = 0.95) # before
```

(continues on next page)

(continued from previous page)

```
##                    2.5 %        97.5 %
## (Intercept)  3.435759991  4.4706529860
## Meducy      -0.018860497  0.0613339120
## Mexp         0.004602597  0.0465899167
## Mexp2       -0.000971303 -0.0001652273
## Mgender     -0.658085143 -0.3281727396

confint(obj = mincer1565A, level = 0.95) # after
##                    2.5 %        97.5 %
## (Intercept)  3.46800378  4.6372462758
## Meducy      -0.03305743  0.0612754964
## Mexp         0.01024867  0.0550935366
## Mexp2       -0.00119162 -0.0003294784
## Mgender     -0.71564602 -0.3630823543
```

If the new estimates fall within the original confidence interval and the new and original confidence intervals are greatly overlapping, the data can be considered valid for this type of regression after anonymization. Fig. 7.2 shows the point estimates and confidence intervals for the gender coefficient in this trade-off for a sample income dataset and several SDC methods and parameters. The red dot and confidence bar (on the top) correspond to the estimates for the untreated data, whereas the other confidence bars correspond to the respective SDC methods and different parameters. The anonymization reduces the number of expected re-identifications in the data (left axis) and the point estimates and confidence intervals vary greatly for the different SDC methods. We would choose a method, which reduces the expected number of identifications, while not changing the gender coefficient and having a great overlap of the confidence interval with the confidence interval estimated from the original data.

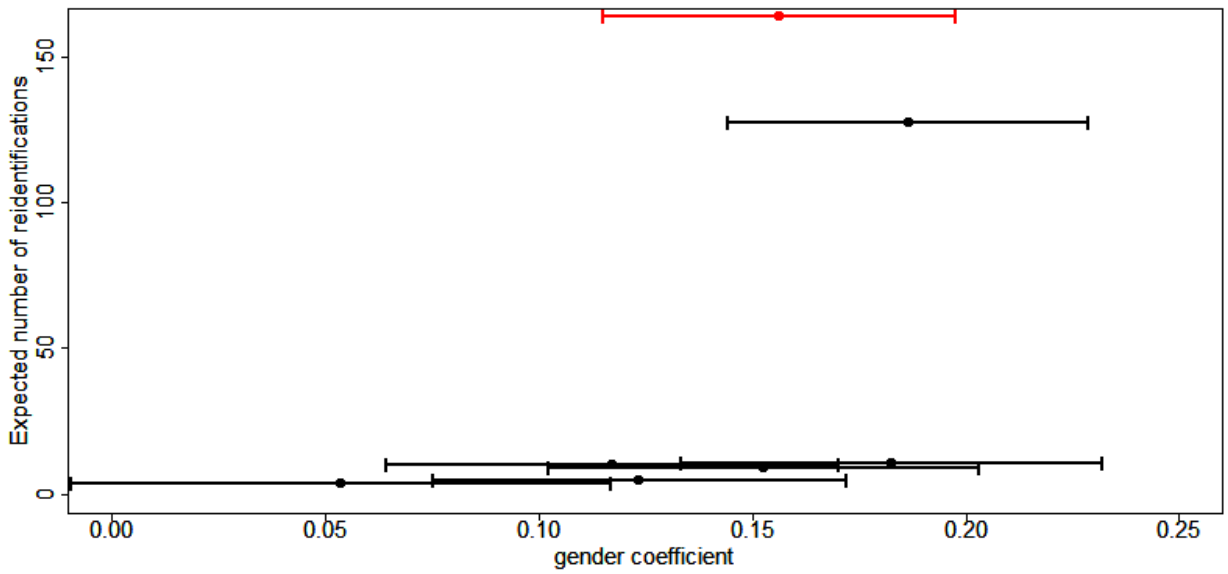

Fig. 7.2: Effect of anonymization on the point estimates and confidence interval of the gender coefficient in the Mincer equation

7.4 Assessing data utility with the help of data visualizations (in *R*)

The use of graphs and other visualization techniques is a good way to assess at a glance how much the data have changed after anonymization, and can aid the selection of appropriate anonymization techniques for the data. Visualizations can be a useful tool to assess the impact on data utility of anonymization methods and helps choose among anonymization methods. The software package *R* provides several functions and packages that can help visualize the results of anonymization. This section lists a few of these functions and packages and provides code examples to illustrate how to implement them. We present the following visualizations:

- histograms and density plots
- boxplots
- mosaic plots

To make appropriate visualizations, we need to use the raw data and the anonymized data. When using an *sdcMicro* object for the anonymization process, the raw data are stored in the "origData" slot of the object and the anonymized variables are in the slots "manipKeyVars", "manipPramVars", "manipNumVars" and "manipStrataVar" slots. See the Section Objects of class sdcMicroObj for more information on *sdcMicro* objects, slots and how to access slots.

7.4.1 Histograms and density plots

Histograms and density plots are useful for quick comparisons of variable distribution before and after anonymization. The advantage of histograms is that the results are exact. Visualization depends on the bin widths and the start point of the first bin, however. Histograms can be used for continuous and semi-continuous variables. Density plots display the kernel density of the data; therefore, the plot depends on the kernel that is chosen and whether the data fits the kernel well. Nevertheless, density plots are a good tool to illustrate the change of values and value ranges of continuous variables.

Histograms can be plotted with function hist() and kernel densities with the functions plot() and density() in *R*. Listing 7.11 provides examples of how to use these functions to illustrate the changes in the variable "INC", an income variable. The function hist() needs as argument the break points for the histogram. The results are shown in Fig. 7.3 and Fig. 7.4. The histograms and density plots give a clear indication how the values have changed: the variability of the data has increased and the shape of the distribution has changed.

Note: The vertical axes of the histograms have different scales.

Listing 7.11: Plotting histograms and kernel densities

```
# Plot histograms
# Plot histogram before anonymization
hist(sdcInitial@origData$INC, breaks = (0:180)*1e2,
     main =  "Histogram income - original data")

# Plot histogram after anonymization (noise addition)
hist(sdcInitial@manipNumVars$INC, breaks = (-20:190)*1e2,
     main = "Histogram income - anonymized data")

# Plot densities
# Plot original density curve
plot(density(sdcInitial@origData$INC), xlim = c(0, 8000), ylim = c*(0, 0.006),
     main = "Density income", xlab = "income")
par (new = TRUE)

```

(continues on next page)

(continued from previous page)

```
# Plot density curve after anonymization (noise addition)
plot(density(sdcInitial@manipNumVars$INC), xlim = c(0, 8000), ylim = c(0, 0.006),
     main = "Density income", xlab = "income")
```

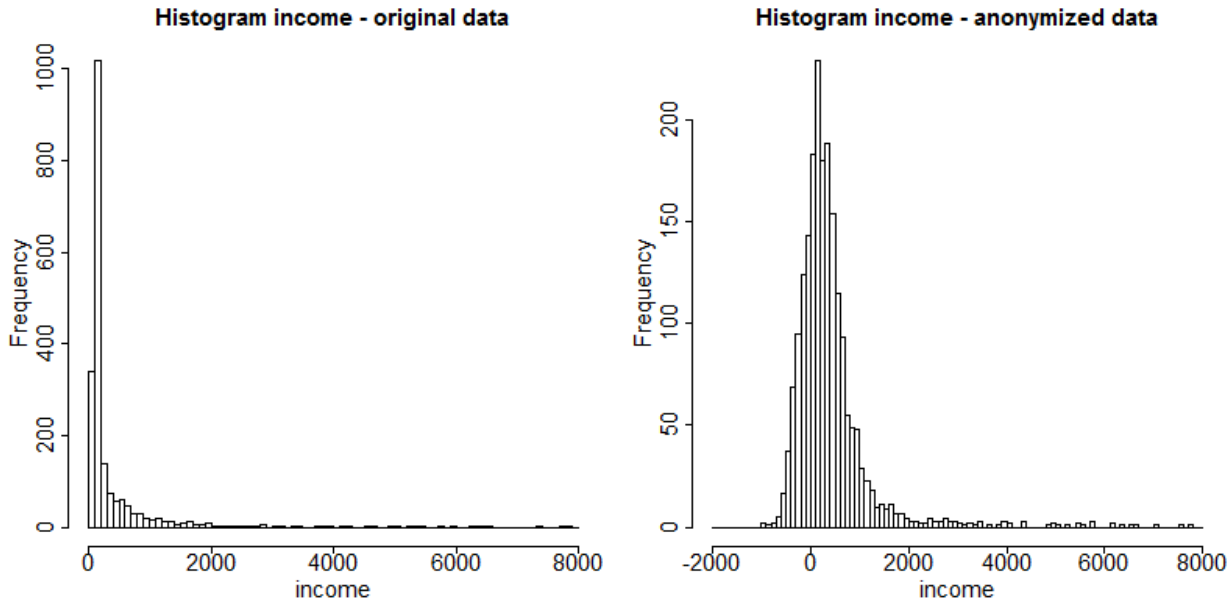

Fig. 7.3: Histograms of income before and after anonymization

7.4.2 Box plots

Box plots give a quick overview of the changes in the spread and outliers of continuous variables before and after anonymization. Listing 7.12 shows how to generate box plots in *R* with the function boxplot(). The result in Fig. 7.5 shows an example for an expenditure variable after adding noise. The box plot shows clearly that the variability in the expenditure variable increased as a result of the anonymization methods applied.

Listing 7.12: Creating boxplots for continuous variables

```
boxplot(sdcObj@origData$TOTFOOD, sdcObj@manipNumVars$TOTFOOD,
        xaxt = 'n', ylab = "Expenditure")
axis(1, at = c(1,2), labels = c('before', 'after'))
```

7.4.3 Mosaic plots

Univariate and multivariate mosaic plots are useful for showing changes in the tabulations of categorical variables, especially when comparing several "scenarios" next to one another. A scenario here refers to the choice of anonymization methods and their parameters. With mosaic plots we can, for instance, quickly see the effect of different levels of k-anonymity or differences in the importance vectors in the local suppression algorithm (see the Section Local suppression).

We illustrate the changes in tabulations with an example of the variable "WATER" before and after applying PRAM. We can use mosaic plots to quickly see the changes for each category. Listing 7.13 shows the code in *R*. The function mosaicplot() is available in base *R*. To plot a tabulation, first the tabulation must be made with the table() function.

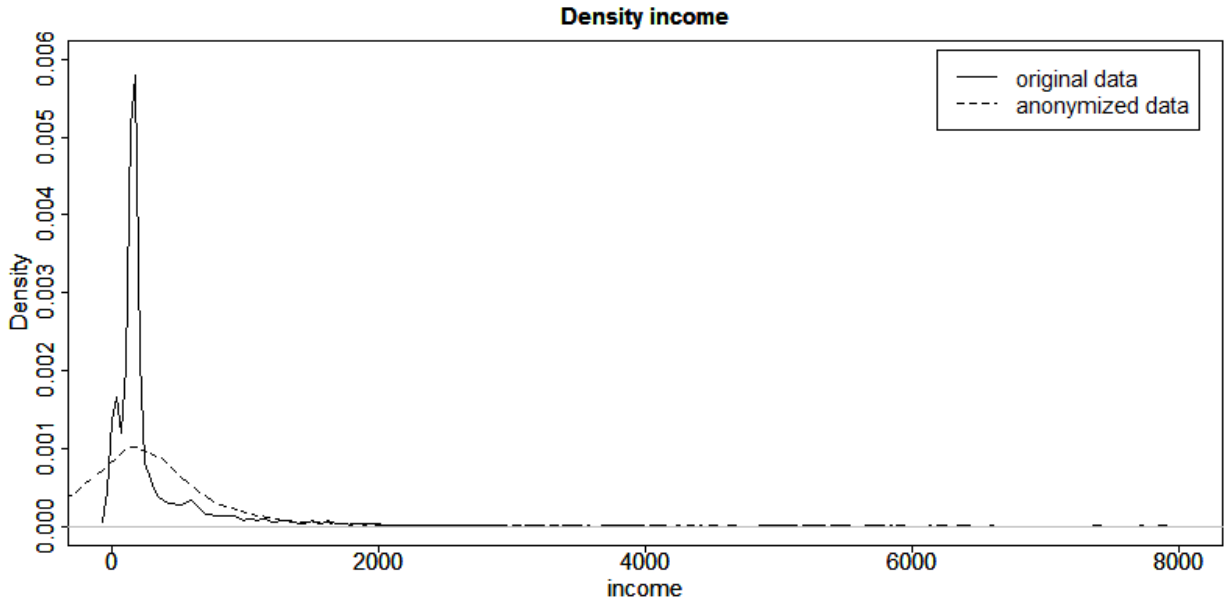

Fig. 7.4: Density plots of income before and after anonymization

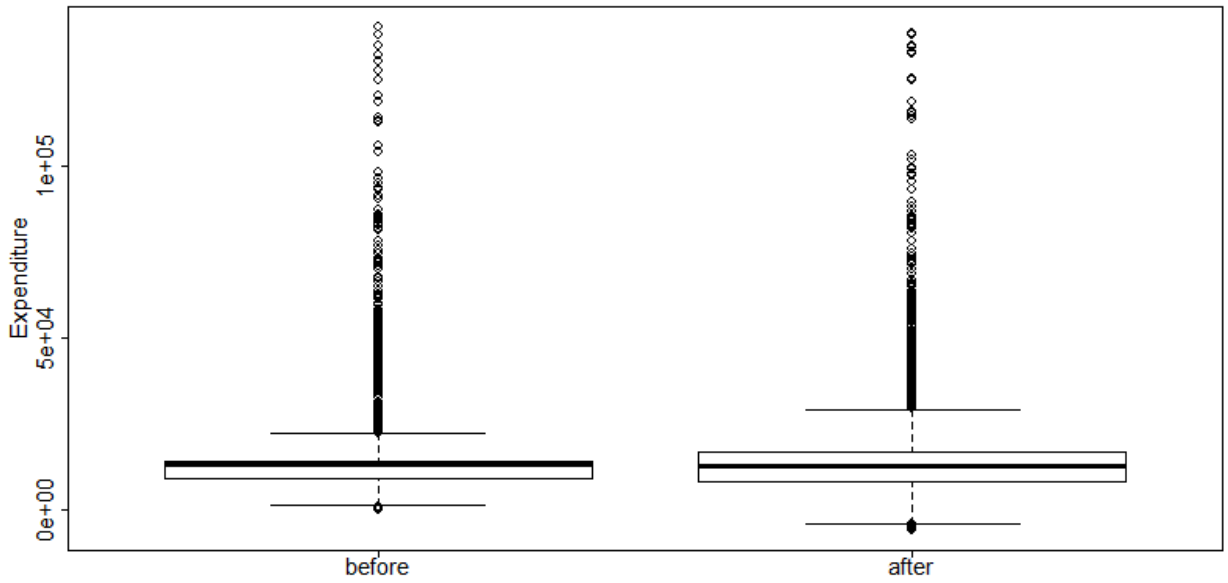

Fig. 7.5: Example of box plots of an expenditure variable before and after anonymization

To show the labels in the mosaicplot(), we change the class of the variables to 'factor' (see the Section Classes in R). Looking at the mosaic plot in Fig. 7.6 we see invariant PRAM has virtually no influence on the univariate distribution.

Listing 7.13: Creating univariate mosaic plots

```
# Collecting data of variable WATER before and after anonymization,
# assigning factor levels for labels in plot
dataWater <- t(cbind(table(factor(sdcHH@origData$WATER,
                                  levels = c(1, 2, 3, 4, 5, 6, 7, 8, 9),
                                  labels = c("Pipe (own tap)", "Public standpipe",
                                             "Borehole", "Wells (protected)",
                                             "Wells (unprotected)", "Surface water",
                                             "Rain water", "Vendor/truck", "Other"))),
                     table(factor(sdcHH@manipPramVars$WATER,
                                  levels = c(1,2, 3, 4, 5, 6, 7, 8, 9),
                                  labels = c("Pipe (own tap)", "Public standpipe",
                                             "Borehole", "Wells (protected)",
                                             "Wells (unprotected)", "Surface water",
                                             "Rain water", "Vendor/truck","Other")))))
rownames(dataWater) <- c("before", "after")

# Plotting mosaic plot
mosaicplot(dataWater, main = "", color = 2:10, las = 2)
```

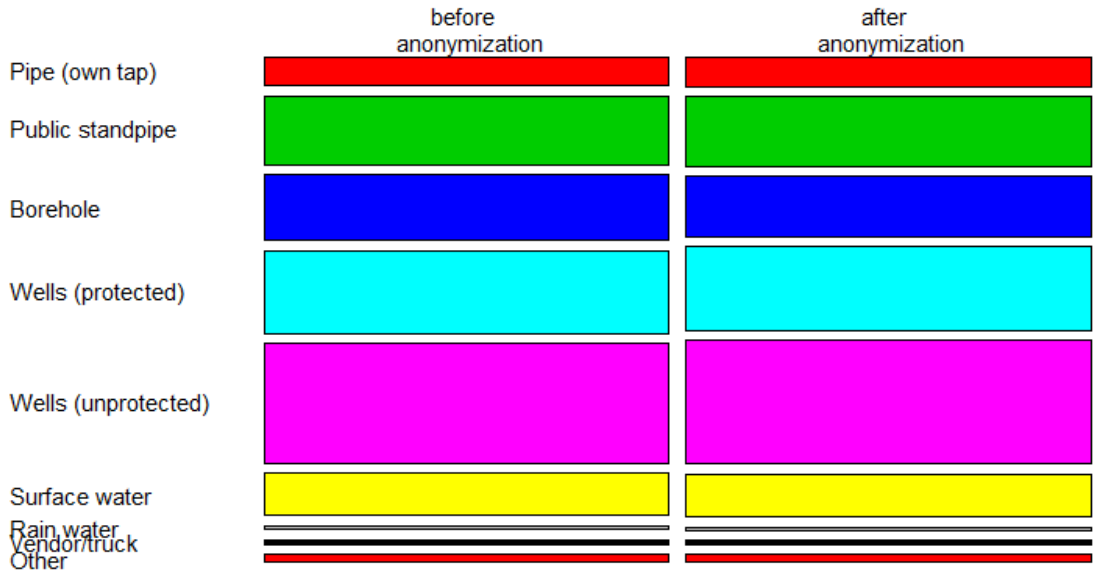

Fig. 7.6: Mosaic plot to illustrate the changes in the WATER variable

We use the variables "gender" and "relationship status" to illustrate the use of mosaic plots for the illustration of changes in univariate tabulations introduced by several sets of anonymization methods. Table 7.1 provides the methods applied in each scenario. Scenario 0, the base scenario, shows the original categories of the gender and relationship status variables, while scenarios 1 to 6 show shifts in the categories after applying different anonymization techniques. Table 6.1 provides a description of the anonymization methods used in each scenario. In total we visualize the impact of six different sets of anonymization methods. We can use mosaic plots to quickly see which set of methods has what impact on the gender and relationship status variables, which can be used to select the best scenario. Looking at the mosaic plots in Fig. 7.7 , we see that scenarios 2, 5 and 6 give the smallest changes for the gender variable and scenarios 3 and 4 for the relationship status variable.

Table 7.1: Description of anonymization methods by scenario

Scenario	Description of anonymization methods applied
0 (base)	Original data, no treatment
1	Recode age (five-year intervals), plus local suppression (required k = 3, high importance on water, toilet and literacy variables)
2	Recode age (five-year intervals), plus local suppression (required k = 5, no importance vector)
3	Recode age (five-year intervals), plus local suppression (required k = 3, high importance on toilet), while also recoding region, urban, education level and occupation variables
4	Recode age (five-year steps), plus local suppression (required k = 5, high importance on water, toilet and literacy), while also recoding region, urban, education level and occupation variables
5	Recode age (five-year intervals), plus local suppression (required k = 3, no importance vector), microaggregation (wealth index), while also recoding region, urban, education level and occupation variables
6	Recode age (five-year intervals) plus local suppression (required k=3, no importance vector), PRAM literacy, while also recoding region, urban, education level and occupation variables

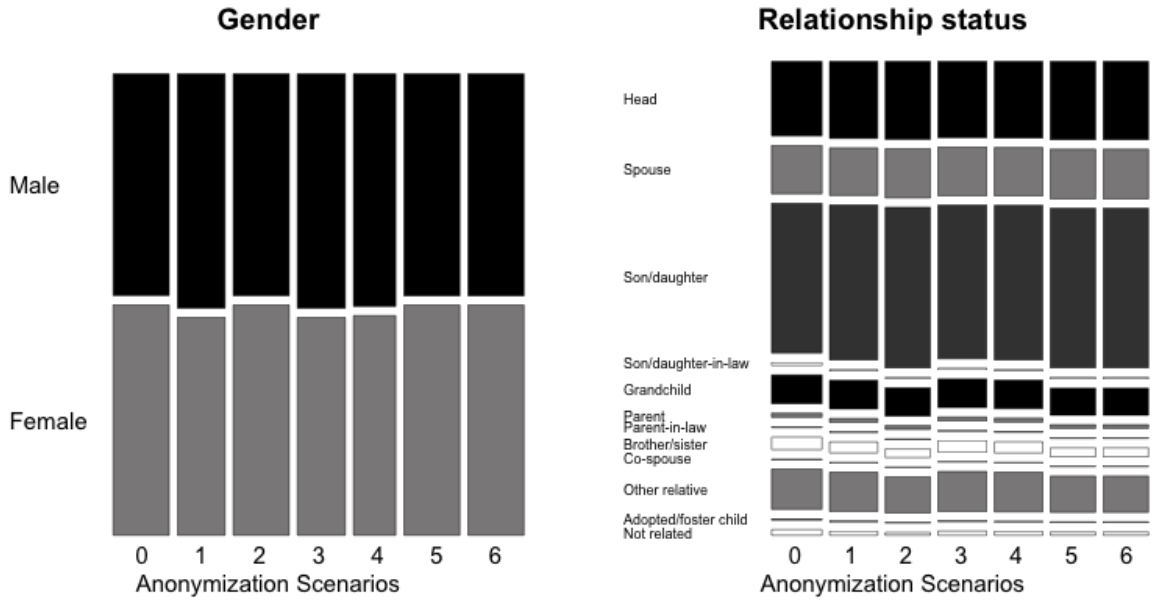

Fig. 7.7: Comparison of treated vs. untreated gender and relationship status variables with mosaic plots

As we discussed in the Section **'PRAM (Post RAndomization Method) <anon_methods.html# PRAM (Post RAndomization Method)'__** , invariant PRAM preserves the univariate distributions. Therefore, in this case it is more interesting to look at the multivariate mosaic plots. Mosaic plots are also a powerful tool to show changes in cross-tabulations/contingency tables. Listing 7.14 shows how to generate mosaic plots for two variables. To compare the changes, we need to compare two different plots. Fig. 7.8 and Fig. 7.9 illustrate that (invariant) PRAM does not preserve the two-way tables in this case.

Listing 7.14: Creating multivariate mosaic plots

```
# Before anonymization: contingency table and mosaic plot
ROOFTOILETbefore <- t(table(factor(sdcHH@origData$ROOF, levels = c(1,2, 3, 4, 5, 9),
                                   labels = c("Concrete/cement/ \n brick/stone", "Wood
↪",
```

(continues on next page)

(continued from previous page)

```
                                                "Bamboo/thatch", "Tiles/shingles",
                                                "Tin/metal sheets", "Other")),
                                  factor(sdcHH@origData$TOILET, levels = c(1,2, 3, 4, 9),
                                         labels = c("Flush \n toilet",
                                                    "Improved \n pit \n latrine",
                                                    "Pit \n latrine", "No \n facility",
                                                    "Other"))))
mosaicplot(ROOFTOILETbefore, main = "", las = 2, color = 2:6)

# After anonymization: contingency table and mosaic plot
ROOFTOILETafter <- t(table(factor(sdcHH@manipPramVars$ROOF, levels = c(1,2, 3, 4, 5,␣
↪9),
                                       labels = c("Concrete/cement/ \n brick/stone", "Wood
↪",
                                                  "Bamboo/thatch", "Tiles/shingles",
                                                  "Tin/metal sheets", "Other")),
                                 factor(sdcHH@manipPramVars$TOILET, levels = c(1,2, 3, 4,␣
↪9),
                                       labels = c("Flush \n toilet",
                                                  "Improved \\n pit \n latrine",
                                                  "Pit \n latrine", "No \n facility",
                                                  "Other"))))
mosaicplot(ROOFTOILETafter, main = "", las = 2, color = 2:6)
```

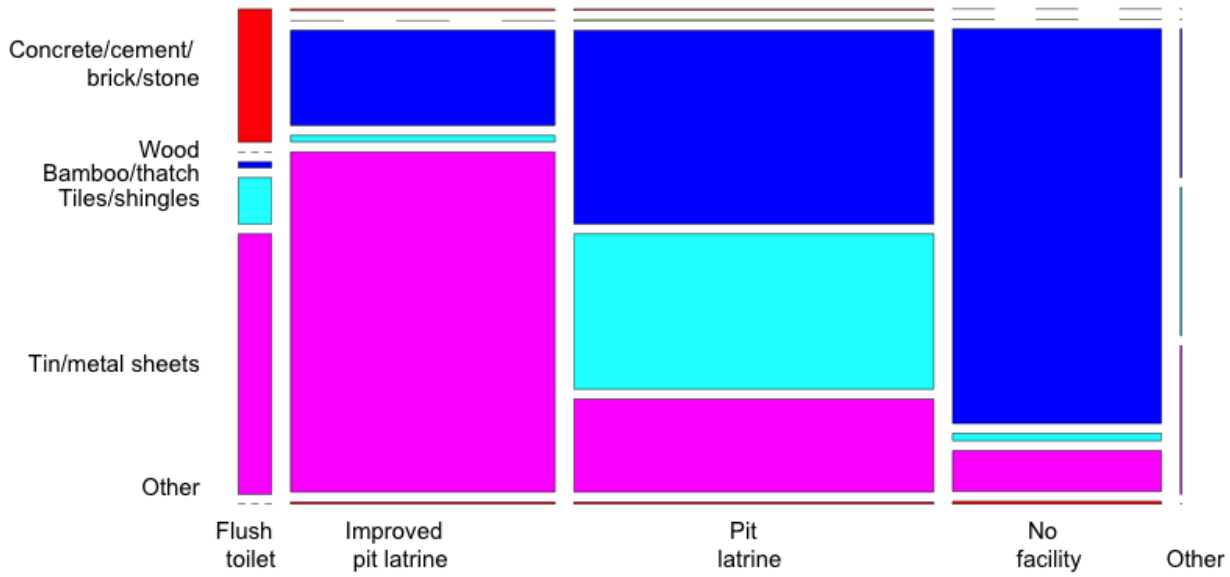

Fig. 7.8: Mosaic plot of the variables ROOF and TOILET before anonymization

7.5 Choice of utility measure

Besides the users' requirements on the data, the utility measures should be chosen in accordance with the variable types and anonymization methods employed. The employed utility measures can be a combination of both general and user-specific measures. As discussed earlier, different utility measures should be used for continuous and categorical data. Furthermore, some utility measures are not informative after certain anonymization methods have been applied. For

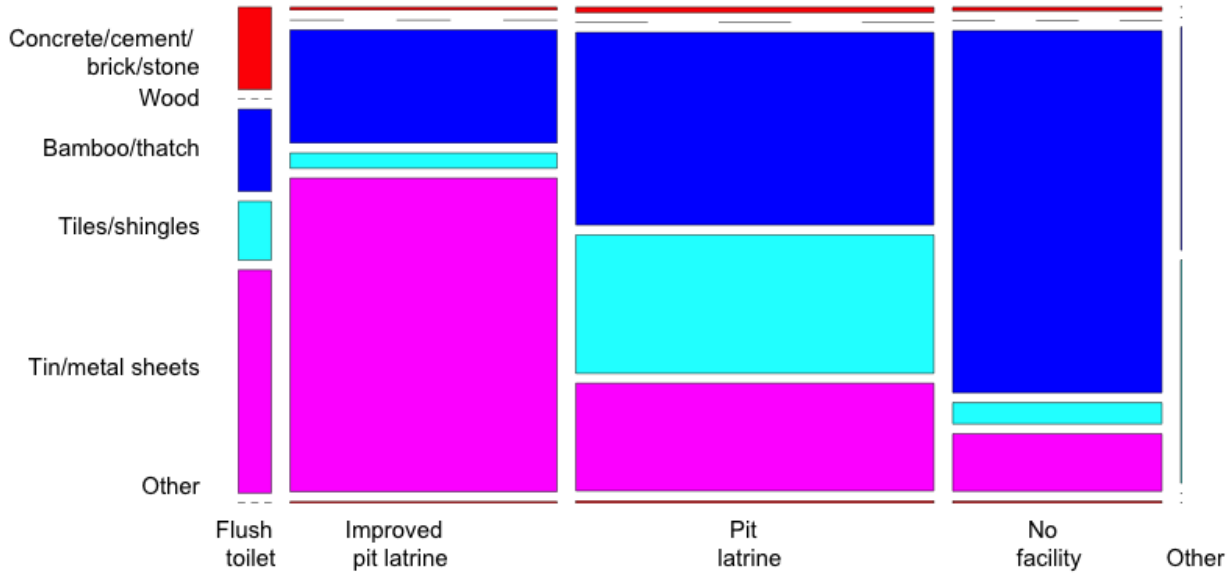

Fig. 7.9: Mosaic plot of the variables ROOF and TOILET after anonymization

example, after applying perturbative methods that interchange data values, comparing values directly is not useful because they will give the impression of high levels of information loss. In such cases, it is more informative to look at means, covariances and benchmarking indicators that can be computed from the data. Furthermore, it is important not only to focus on the characteristics of variables one by one, but also on the interactions between variables. This can be done by cross-tabulations and regressions. In general, when anonymizing sampled data, it is advisable to compute confidence intervals around estimates to interpret the magnitude of changes.

Recommended Reading Material on Measuring Utility and Information Loss

A.G. De Waal and L.C.R.J. Willenborg. 1999. "Information Loss through Global Recoding and Local Suppression" In Netherlands Official Statistics: Special Issue on SDC, 14, 17-10.

J. Domingo-Ferrer, J.M. Mateo-Sanz and V. Torra. 2001. "Comparing SDC Methods for Microdata on the basis of Information Loss and Disclosure Risk". In Pre-proceedings of ETK-NTTS 2001 (vol. 2), 807-826. http://neon.vb.cbs.nl/casc/NTTSJosep.pdf

J. Domingo-Ferrer and V. Torra. 2001. "Disclosure Protection Methods and Information Loss for Microdata". In P. Doyle, J.I. Lane, J.J.M. Theeuwes and L. Zayatz (eds.) *Theory and Practical Applications for Statistical Agencies*, 91-110, Amsterdam. http://crises-deim.urv.cat/webCrises/publications/bcpi/cliatpasa01Disclosure.pdf

---

References

CHAPTER

EIGHT

SDC WITH *SDCMICRO* IN *R*: SETTING UP YOUR DATA AND MORE

8.1 Installing *R*, *sdcMicro* and other packages

This guide is based on the software package *sdcMicro*, which is an add-on package for the statistical software *R*. Both *R* and *sdcMicro*, as well as other *R* packages, are freely available from the CRAN (Comprehensive R Archive Network) website for Linux, Mac and Windows (http://cran.r-project.org). This website also offers descriptions of packages. Besides the standard version of *R*, there is a more user-friendly user interface for *R*: *RStudio*. *RStudio* is also freely available for Linux, Mac and Windows (http://www.rstudio.com). The *sdcMicro* package is dependent on (i.e., uses) other *R* packages that must be installed on your computer before using *sdcMicro*. Those will automatically be installed when installing *sdcMicro*. For some functionalities, we use still other packages (such as *foreign* for reading data and some graphical packages). If so, this is indicated in the appropriate section in this guide. *R*, *RStudio*, the *sdcMicro* package and its dependencies and other packages have regular updates. It is strongly recommended to regularly check for updates: this requires installing a new version for an update of *R;* with the update.packages() command or using the menu options in *R* or *RStudio* one can update the installed packages.

When starting *R* or *RStudio*, it is necessary to specify each time which packages are being used by loading those. This loading of packages can be done either with the library() or the require() function. Both options are illustrated in Listing 8.1.

Listing 8.1: Loading required packages

```
1 library(sdcMicro) # loading the sdcMicro package
2 require(sdcMicro) # loading the sdcMicro package
```

All packages and functions are documented. The easiest way to access the documentation of a specific function is to use the built-in help, which generally gives an overview of the parameters of the functions as well as some examples. The help of a specific function can be called by a question mark followed by the function name without any arguments. Listing 8.2 shows how to call the help file for the microaggregation() function of the *sdcMicro* package.[1] The download page of the each package on the CRAN website also provides a reference manual with a complete overview of the functions in the package.

[1] Often it is also useful to search the internet for help on specific functions in *R*. There are many fora where *R* users discuss issues they encounter. One particularly useful site is stackoverflow.com.

Listing 8.2: Displaying help for functions

```
?microaggregation # help for microaggregation function
```

When issues or bugs in the *sdcMicro* package are encountered, comments, remarks or suggestions can be posted for the developers of *sdcMicro* on their GitHub.

8.2 Read functions in *R*

The first step in the SDC process when using *sdcMicro* is to read the data into *R* and create a dataframe.[2] *R* is compatible with most statistical data formats and provides read functions for most types of data. For those read functions, it is sometimes necessary to install additional packages and their dependencies in *R*. An overview of data formats, functions and the packages containing these functions is provided in Table 8.1. These functions are also available as write (e.g., write_dta()) to save the anonymized data in the required format.[3]

Table 8.1: Packages and functions for reading data in *R*

Type/software	Extension	Package	Function
SPSS	.sav	haven	read_sav()
STATA (v. 5-14)	.dta	haven	read_dta()
SAS	.sas7bdat	haven	read_sas()
Excel	.csv	utils (base package)	read.csv()
Excel	.xls/.xlsx	readxl	readxl()

Most of these functions have options that specify how to handle missing values and variables with factor levels and value labels. Listing 8.3, Listing 8.4 and Listing 8.5 provide example code for reading in a *STATA* (.dta) file, an *Excel* (.csv) file and a *SPSS* (.sav) file.

Listing 8.3: Reading in a *STATA* file

```
setwd("/Users/World Bank") # working directory with data file
fname = "data.dta" # name of data file
library(haven) # loads required package for read/write function for STATA files
file <- read_dta(fname)
# reads the data into the data frame tbl called file
```

Listing 8.4: Reading in a *Excel* file

```
setwd("/Users/World Bank") # working directory with data file
fname = "data.csv" # name of data file
file <- read.csv(fname, header = TRUE, sep = ",", dec = ".")
# reads the data into the data frame called file,
# the first line contains the variable names,
# fields are separated with commas, decimal points are indicated with '.'
```

Listing 8.5: Reading in a *SPSS* file

```
setwd("/Users/World Bank") # working directory with data file
fname = "data.sav" # name of data file
```

(continues on next page)

[2] A dataframe is an object class in *R*, which is similar to a data table or matrix.

[3] Not all functions are compatible with all versions of the respective software package. We refer to the help files of the read and write functions for more information.

(continued from previous page)

```
library(haven) # loads required package for read/write function for SPSS files
file <- read_sav(fname)
# reads the data into the data frame called file
```

The maximum data size in *R* is technically restricted. The maximum size depends on the *R* build (32-bit or 64-bit) and the operating system. Some SDC methods require long computation times for large datasets (see the Section on *Computation time*).

8.3 Missing values

The standard way missing values are represented in *R* is by the symbol 'NA', which is different to impossible values, such as division by zero or the log of a negative number, which are represented by the symbol 'NaN'. The value 'NA' is used for both numeric and categorical variables.[4] Values suppressed by the localSuppression() routine are also replaced by the 'NA' symbol. Some datasets and statistical software might use different values for missing values, such as '999' or strings. It is possible to include arguments in read functions to specify how missing values in the dataset should be treated and automatically recode missing values to 'NA'. For instance, the function read.table() has the 'na.strings' argument, which replaces the specified strings with 'NA' values.

Missing values can also be recoded after reading the data into *R*. This may be necessary if there are several different missing value codes in the data, different missing value codes for different variables or the read function for the datatype does not allow specifying the missing value codes. When preparing data, it is important to recode any missing values that are not coded as 'NA' to 'NA' in *R* before starting the anonymization process to ensure the correct measurement of risk (e.g., k-anonymity), as well as to ensure that many of the methods are correctly applied to the data. Listing 8.6 shows how to recode the value '99' to 'NA' for the variable "toilet".

Listing 8.6: Recoding missing values to NA

```
file[file[,'toilet'] == 99,'toilet'] <- NA
# Recode missing value code 99 to NA for variable toilet
```

8.4 Classes in R

All objects in *R* are of a specific class, such as integer, character, matrix, factor or dataframe. The class of an object is an attribute from which the object inherits. To find out the class of an object, one can use the function class(). Functions in *R* might require objects or arguments of certain classes or functions might have different functionality depending on the class of the argument. Examples are the write functions that require dataframes and most functions in the *sdcMicro* package that require either dataframes or *sdcMicro* objects. The functionality of the functions in the *sdcMicro* package differs for dataframes and *sdcMicro* objects. It is easy to change the class attribute of an object with functions that start with "as.", followed by the name of the class (e.g., as.factor(), as.matrix(), as.data.frame()). Listing 8.7 shows how to check the class of an object and change the class to "data.frame". Before changing the class attribute of the object "file", it was in the class "matrix". An important class defined and used in the *sdcMicro* package is the class named *sdcMicroObj*. This class is described in the next section.

Listing 8.7: Changing the class of an object in *R*

```
# Finding out the class of the object 'file'
class(file)
"matrix"

```

(continues on next page)

[4] This is regardless of the class of the variable in *R*. See the Section *Classes in R* for more on classes in *R*.

(continued from previous page)

```
# Changing the class to data frame
file <- as.data.frame(file)

# Checking the result class(file)
"data.frame"
```

8.5 Objects of class *sdcMicroObj*

The *sdcMicro* package is built around objects[5] of class *sdcMicroObj*, a class especially defined for the *sdcMicro* package. Each member of this class has a certain structure with slots that contain information regarding the anonymization process (see Table 8.2 for a description of all slots). Before evaluating risk and utility and applying SDC methods, creating an object of class *sdcMicro* is recommended. All examples in this guide are based on these objects. The function used to create an *sdcMicro* object is createSdcObj(). Most functions in the *sdcMicro* package, such as microaggregation() or localSuppression(), automatically use the required information (e.g., quasi-identifiers, sample weights) from the *sdcMicro* object if applied to an object of class *sdcMicro*.

The arguments of the function createSdcObj() allow one to specify the original data file and categorize the variables in this data file before the start of the anonymization process.

Note: For this, disclosure scenarios must already have been evaluated and quasi-identifiers selected. In addition, one must ensure there are no problems with the data, such as variables containing only missing values.

In Listing 8.8, we show all arguments of the function createSdcObj(), and first define vectors with the names of the different variables. This practice gives a better overview and later allows for quick changes in the variable choices if required. We choose the categorical quasi-identifiers (keyVars); the variables linked to the categorical quasi-identifiers that need the same suppression pattern (ghostVars, see the Section Local suppression); the numerical quasi-identifiers (numVars); the variables selected for applying PRAM (pramVars); a variable with sampling weights (weightVar); the clustering ID (hhId, e.g., a household ID, see the Section Household risk); a variable specifying the strata (strataVar) and the sensitive variables specified for the computation of *l*-diversity (sensibleVar , see the Section l-diversity).

Note: Most SDC methods in the sdcMicro package are automatically applied within the strata, if the 'strataVar' argument is specified.

Examples are local suppression and PRAM. Not all variables must be specified, e.g., if there is no hierarchical (household) structure, the argument 'hhId' can be omitted. The names of the variables correspond to the names of the variables in the dataframe containing the microdata to be anonymized. The selection of variables is important for the risk measures that are automatically calculated. Furthermore, several methods are by default applied to all variables of one sort, e.g., microaggregation to all key variables.[6] After selecting these variables, we can create the *sdcMicro* object. To obtain a summary of the object, it is sufficient to write the name of the object.

Listing 8.8: Selecting variables and creating an object of class *sdcMicroObj* for the SDC process in *R*

```
# Select variables for creating sdcMicro object
# All variable names should correspond to the names in the data file
# selected categorical key variables
selectedKeyVars = c('region', 'age', 'gender', 'marital', 'empstat')
```

(continues on next page)

[5] Class *sdcMicroObj* has S4 objects, which have slots or attributes and allow for object-oriented programming.
[6] Unless otherwise specified in the arguments of the function.

(continued from previous page)

```
# selected linked variables (ghost variables)
selectedGhostVars = c('urbrur')

# selected categorical numerical variables
selectedNumVar = c('wage', 'savings')

# weight variable
selectedWeightVar = c('wgt')

# selected pram variables
selectedPramVars = c('roof', 'wall')

# household id variable (cluster)
selectedHouseholdID = c('idh')

# stratification variable
selectedStrataVar = c('strata')

# sensitive variables for l-diversity computation
selectedSensibleVar = c('health')

# creating the sdcMicro object with the assigned variables
sdcInitial <- createSdcObj(dat         = file,
                                          keyVars     = selectedKeyVars,
                                      ghostVars   = selectedGhostVars,
                                      numVar      = selectedNumVar,
                                      weightVar   = selectedWeightVar,
                                      pramVars    = selectedPramVars,
                                      hhId        = selectedHouseholdID,
                           strataVar   = selectedStrataVar,
                           sensibleVar = selectedSensibleVar)

# Summary of object
sdcInitial

## Data set with 4580 rows and 14 columns.
## --> Categorical key variables: region, age, gender, marital, empstat
## --> Numerical key variables: wage, savings
## --> Weight variable: wgt
## ----------------------------------------------------------------------------
##
## Information on categorical Key-Variables:
##
## Reported is the number, mean size and size of the smallest category for recoded␣
↪variables.
## In parenthesis, the same statistics are shown for the unmodified data.
## Note: NA (missings) are counted as seperate categories!
##
## Key Variable Number of categories Mean size
## region 2 (2) 2290.000 (2290.000)
## age 5 (5) 916.000 (916.000)
## gender 3 (3) 1526.667 (1526.667)
## marital 8 (8) 572.500 (572.500)
## empstat 3 (3) 1526.667 (1526.667)
##
## Size of smallest
```

(continues on next page)

(continued from previous page)

```
## 646 (646)
## 16 (16)
## 50 (50)
## 26 (26)
## 107 (107)
## ------------------------------------------------------------------------
##
## Infos on 2/3-Anonymity:
##
## Number of observations violating
## - 2-anonymity: 157
## - 3-anonymity: 281
##
## Percentage of observations violating
## - 2-anonymity: 3.428 %
## - 3-anonymity: 6.135 %
## ------------------------------------------------------------------------
##
## Numerical key variables: wage, savings
##
## Disclosure risk is currently between [0.00%; 100.00]
##
## Current Information Loss:
## IL1: 0.00
## Difference of Eigenvalues: 0.000%
## ------------------------------------------------------------------------
```

Table 8.2 presents the names of the slots and their respective contents. The slot names can be listed using the function slotNames(), which is illustrated in Listing 8.9. Not all slots are used in all cases. Some slots are filled only after applying certain methods, e.g., evaluating a specific risk measure. Certain slots of the objects can be accessed by accessor functions (e.g., extractManipData for extracting the anonymized data) or print functions (e.g., print()) with the appropriate arguments. The content of a slot can also be accessed directly with the '@' operator and the slot name. This is illustrated for the risk slot in Listing 8.9. This functionality can be practical to save intermediate results and compare the outcomes of different methods. Also, for manual changes to the data during the SDC process, such as changing missing value codes or manual recoding, the direct accession of the data in the slots with the manipulated data (i.e., slot names starting with 'manip') is useful. Within each slot there are generally several elements. Their names can be shown with the names() function and they can be accessed with the '$' operator. This is shown for the element with the individual risk in the risk slot.

Listing 8.9: Displaying slot names and accessing slots of an S4 object

```
# List names of all slots of sdcMicro object
slotNames(sdcInitial)

##  [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"

# Accessing the risk slot
sdcInitial@risk

```

(continues on next page)

(continued from previous page)

```
# List names within the risk slot
names(sdcInitial@risk)
## [1] "global"  "individual"  "numeric"

# Two ways to access the individual risk within the risk slot
sdcInitial@risk$individual
get.sdcMicroObj(sdcInitial, "risk")$individual
```

Table 8.2: Slot names and slot description of *sdcMicro* object

Slotname	Content
origData	original data as specified in the dat argument of the createSdcObj() function
keyVars	indices of columns in origData with specified categorical key variables
pramVars	indices of columns in origData with specified PRAM variables
numVars	indices of columns in origData with specified numerical key variables
ghostVars	indices of columns in origData with specified ghostVars
weightVar	indices of columns in origData with specified weight variable
hhId	indices of columns in origData with specified cluster variable
strataVar	indices of columns in origData with specified strata variable
sensibleVar	indices of columns in origData with specified sensitive variables for lDiversity
manipKeyVars	manipulated categorical key variables after applying SDC methods (cf. keyVars slot)
manipPramVars	manipulated PRAM variables after applying PRAM (cf. pramVars slot)
manipNumVars	manipulated numerical key variables after applying SDC methods (cf. numVars slot)
manipGhostVars	manipulated ghost variables (cf. ghostVars slot)
manipStrataVar	manipulated strata variables (cf. strataVar slot)
originalRisk	global and individual risk measures before anonymization
risk	global and individual risk measures after applied SDC methods
utility	utility measures (il1 and eigen)
pram	details on PRAM after applying PRAM
localSuppression	number of suppression per variable after local suppression
options	options specified
additionalResults	additional results
set	list of slots currently in use (for internal use)
prev	information to undo one step with the undo() function
deletedVars	variables deleted (direct identifiers)

There are two options to save the results after applying SDC methods:

- Overwriting the existing *sdcMicro* object, or
- Creating a new *sdcMicro* object. The original object will not be altered and can be used for comparing results. This is especially useful for comparing several methods and selecting the best option.

In both cases, the result of any function has to be re-assigned to an object with the '<-' operator. Both methods are illustrated in Listing 8.10.

Listing 8.10: Saving results of applying SDC methods

```
# Applying local suppression and reassigning the results to the same sdcMicro object
sdcInitial <- localSuppression(sdcInitial)

# Applying local suppression and assigning the results to a new sdcMicro object
sdc1 <- localSuppression(sdcInitial)
```

If the results are reassigned to the same *sdcMicro* object, it is possible to undo the last step in the SDC process. This is useful when changing parameters. The results of the last step, however, are lost after undoing that step.

Note: The undolast() function can be used to go only one step back, not several.

The result must also be reassigned to the same object. This is illustrated in Listing 8.11.

Listing 8.11: Undo last step in SDC process

```
# Undo last step in SDC process
sdcInitial <- undolast(sdcInitial)
```

8.6 Household structure

If the data has a hierarchical structure and some variables are measured on the higher hierarchical level and others on the lower level, the SDC process should be adapted accordingly (see also the Sections Household risk and Anonymization of the quasi-identifier household size). A common example in social survey data is datasets with a household structure. Variables that are measured on the household level are, for example, household income, type of house and region. Variables measured on the individual level are, for example, age, education level and marital status. Some variables are measured on the individual level, but are nonetheless the same for all household members in almost all households. These variables should be treated as measured on the household level from the SDC perspective. An example is the variable religion for some countries.

The SDC process should be divided into two stages in cases where the data have a household structure. First, the variables on the higher (household) level should be anonymized; subsequently, the treated higher-level variables should be merged with the individual variables and anonymized jointly. In this section, we explain how to extract household variables from a file and merge them with the individual levels variables after treatment in *R*. We illustrate this process with an example of household and individual-level variables.

These steps are illustrated in Listing 8.12. We require both an individual ID and a household ID in the dataset; if they are lacking, they must be generated. The individual ID has to be unique for every individual in the dataset and the household ID has to be unique across households. The first step is to extract the household variables and save them in a new dataframe. We specify the variables that are measured at the household level in the string vector "HHVars" and subtract only these variables from the dataset. This dataframe will have for each household the same number of entries as it has household members (e.g., if a household has four members, this household will appear four times in the file). We next apply the function unique() to select only one record per household. This argument of the unique function is the household ID, which is the same for all household members, but unique across households.

Listing 8.12: Create a household level file with unique records (remove duplicates)

```
# Create subset of file with only variables measured at household level
HHVars <- c('region', 'hhincome')
fileHH <- file[,HHVars]
```

(continues on next page)

(continued from previous page)

```

# Remove duplicated rows based on the household ID / only every household once in␣
↪fileHH
fileHH <- unique(fileHH, by = c('HID'))

# Dimensions of fileHH (number of households)
dim(fileHH)
```

After anonymizing the household variables based on the dataframe "fileHH", we recombine the anonymized household variables with the original variables, which are measured on the individual level. We can extract the individual-level variables from the original dataset using "INDVars" – a string vector with the individual-level variable names. For extracting the anonymized data from the *sdcMicro* object, we can use the extractManipData() function from the *sdcMicro* package. Next, we merge the data using the merge function. The 'by' argument in the merge function specifies the variable used for merging – in this case the household ID, which has the same variable name in both datasets. All other variables should have different names in both datasets. These steps are illustrated in Listing 8.13.

Listing 8.13: Merging anonymized household-level variables with individual-level variables

```
# Extract manipulated household level variables from the SDC object
HHmanip <- extractManipData(sdcHH)

# Create subset of file with only variables measured at individual level
fileIND <- file[,INDVars]

# Merge the file by using the household ID
fileCombined <- merge(HHmanip, fileIND, by = c('HID'))
```

The file *fileCombined* is used for the SDC process with the entire dataset. How to deal with data with household structure is illustrated in the case studies in the Section Case studies.

As discussed in the Section Anonymization of the quasi-identifier household size), the size of a household can also be a quasi-identifier, even if the household size is not included in the dataset as variable. For the purpose of evaluating the disclosure risk, it might be necessary to create such a variable by a headcount of the members of each household. Listing 8.14 shows how to generate a variable household size with values for each individual based on the household ID (HID). Two cases are shown: 1) the file sorted by household ID and 2) the file not sorted.

Listing 8.14: Generating the variable household size

```
# Sorted by HID
file$hhsize <- rep(unname(table(file$HID)), unname(table(file$HID)))

# Unsorted
file$hhsize <- rep(diff(c(1, 1 + which(diff(file$HID) != 0), length(file$HID) + 1)),
                   diff(c(1, 1 + which(diff(file$HID) != 0), length(file$HID) + 1)))
```

Note: In some cases, the order of the individuals within the households can provide information that could lead to re-identification.

An example is information on the relation to the household head. In many countries, the first individual in the household is the household head, the second the partner of the household head and the next few are children. Therefore, the line number within the household could correlate well with a variable that contains information on the relation to the household head. One way to avoid this unintended release of information is to change the order of the individuals within each household at random. Listing 8.15 illustrates a way to do this in *R*.

Listing 8.15: Changing the order of individuals within households

```
# List of household sizes by household
hhsize <- diff(c(1, 1 + which(diff(file$HID) != 0), length(file$HID) + 1))

# Line numbers randomly assigned within each household
set.seed(123)
dataAnon$INDID <- unlist(lapply(hhsize,
                                function(n){sample(1:n, n, replace = FALSE,
                                                   prob = rep(1/n, n))}))

# Order the file by HID and randomized INDID (line number)
dataAnon <- dataAnon[order(dataAnon$HID, dataAnon$INDID),]
```

8.7 Randomizing order and numbering of individuals or households

Often the order and numbering of individuals, households, and also geographical units contains information that could be used by an intruder to re-identify records. For example, households with IDs that are close to one another in the dataset are likely to be geographically close as well. This is often the case in a census, but also in a household survey households close to one another in the dataset likely share the same low level geographical unit if the dataset is sorted in that way. Another example is a dataset that is alphabetically sorted by name. Here, removing the direct identifier name before release is not sufficient to guarantee that the name information cannot be used (e.g. first record has a name which likely starts with 'a'). Therefore, it is often recommended to randomize the order of records in a dataset before release. Randomization can also be done within subsets of the dataset, e.g., within regions. If suppressions were made in the geographical variable used for creating the subsets, randomization within the geographical subsets implies that the geographical variable is the same for all records in the subset and the suppressed value can be easily derived (for instance, in cases where the geographical unit is included in the randomized ID). Therefore, if the variable used for the subsets has suppressed values, randomization should be done at the dataset level and not at the subset level.

Table 8.3 illustrates the need and process of randomizing the order of records in a dataset. The first three columns in Table 8.3 show the original dataset. Some suppressions were made in the variable "district", as shown in columns 4 to 6 ('NA' values). This dataset also already shows the randomized household IDs. The order of the records in the columns 1-3 and columns 4-6 is unchanged. By the order of the records, it is easy to guess the values of the two suppressed values. Both the record before and after have the same value for district as the suppressed values, respectively 3 and 5. After reordering the dataset based on the randomized household IDs, we see that it becomes impossible to reconstruct the suppressed values based on the values of the neighboring records. Note that in this example the randomization was carried out within the regions and the region number is included in the household ID (first digit).

Table 8.3: Illustration of randomizing order of records in a dataset

Original dataset			Dataset with randomized household ID			Dataset for release ordered by the new randomized household ID		
Household ID	Region	District	Randomized household ID	Region	District	Randomized household ID	Region	District
101	1	1	108	1	1	101	1	4
102	1	1	106	1	1	102	1	3
103	1	2	104	1	2	103	1	5
104	1	2	112	1	2	104	1	2
105	1	2	105	1	2	105	1	2
106	1	3	102	1	3	106	1	1
107	1	3	109	1	NA	107	1	3
108	1	3	107	1	3	108	1	1
109	1	4	101	1	4	109	1	NA
110	1	5	111	1	5	110	1	NA
111	1	5	110	1	NA	111	1	5
112	1	5	103	1	5	112	1	2
201	2	6	203	2	6	201	2	6
202	2	6	204	2	6	202	2	6
203	2	6	201	2	6	203	2	6
204	2	6	202	2	6	204	2	6

The randomization is easiest if done before or after the anonymization process with *sdcMicro* and directly on the dataset (data.frame in *R*). To randomize the order, we need an ID, such as an individual ID, household ID or geographical ID. If the dataset does not contain such ID, this should be created first. Listing 8.16 shows how to randomize households. "HID" is the household ID and "regionid" is the region ID. First the variable "HID" is replaced by a randomized variable "HIDrandom". Then the file is sorted by region and the randomized ID and the actual order of the records in the dataset is changed. To make the randomization reproducible, it is advisable to set a seed for the random number generator.

Listing 8.16: Randomize order of households

```
n <- length(file$HID) # number of households

set.seed(123) # set seed
# generate random HID
file$HIDrandom <- sample(1:n, n, replace = FALSE, prob = rep(1/n, n))

# sort file by regionid and random HID
file <- file1[order(file$regionid, file$HIDrandom),]

# renumber the households in randomized order to 1-n
file$HIDrandom <- 1:n
```

8.8 Computation time

Some SDC methods can take a very long time to evaluate in terms of computation. For instance, local suppression with the function localSuppression() of the *sdcMicro* package in *R* can take days to execute on large datasets of more than 30,000 individuals that have many categorical quasi-identifiers. Our experiments reveal that computation time is a function of the following factors: the applied SDC method; data size, i.e., number of observations, number of variables and the number of categories or factor levels of each categorical variable; data complexity (e.g., the number of different combinations of values of key variables in the data); as well as the computer/server specifications.

Table 8.4 gives some indication of computation times for different methods on datasets of different size and complexity based on findings from our experiments. The selected quasi-identifiers and categories for those variables in Table 8.4 are the same in both datasets being compared. Because it is impossible to predict the exact computation time, this table should be used to illustrate how long computations may take. These methods have been executed on a powerful server. Given long computation times for some methods, it is recommended, where possible, to first test the SDC methods on a subset or sample of the microdata, and then choose the appropriate SDC methods. *R* provides functions to select subsets from a dataset. After setting up the code, it can then be run on the entire dataset on a powerful computer or server.

Table 8.4: Computation times of different methods on datasets of different sizes

Dataset with 5,000 observations		Dataset with 45,000 obervations	
Methods	Computation time (hours)	Methods	Computation time (hours)
Top coding age, local suppression (k=3)	11	Top coding age, local suppression (k=3)	268
Recoding age, local suppression (k=3)	8	Recoding age, local suppression (k=3)	143
Recoding age, local suppression (k=5)	10	Recoding age, local suppression (k=5)	156

The number of categories and the product of the number of categories of all categorical quasi-identifiers give an idea of the number of potential combinations (keys). This is only an indication of the actual number of combinations, which influences the computation time to compute, for example, the frequencies of each key in the dataset. If there are many categories but not so many combinations (e.g., when the variables correlate), the computation time will be shorter.

Table 8.5 shows the number of categories for seven datasets with the same variables but of different complexities that were all processed using the same script on 16 processors, in order of execution time. The table also shows an approximation of the number of unique combinations of quasi-identifiers, as indicated by the percentage of observa-

tions violating k-anonymity in each dataset pre-anonymization in relation to processing time. The results in the table clearly indicate that both the number of observations (i.e., sample size) and the complexity of the data play a role in the execution time. Also, using the same script (and hence anonymization methods), the execution time can vary greatly; the longest running time is about 10 times longer than the shortest. Computer specifications also influence the computation time. This includes the processor, RAM and storage media.

Table 8.5: Number of categories (complexity), record uniqueness and computation times

Sample size	Number of categories per quasi-identifier (complexity)						Percentage of observations violating k-anonimity before before anonymization		Execution time in hours
n	Water	Toilet	Occupation	Religion	Ethnicity	Region	k3	k5	
20,014	10	4	70	5	7	6	74	88	53.72
66,285	15	6	39	4	0	24	40	49	67.19
60,747	13	6	70	8	9	4	35	45	74.47
26,601	19	6	84	10	10	10	77	87	108.84
38,089	17	6	30	5	56	9	70	81	198.90
35,820	19	7	67	6	NA	6	81	90	267.60
51,976	12	6	32	8	50	12	77	87	503.58

The large-scale experiment executed for this guide utilized 75 microdata files from 52 countries, using surveys on topics including health, labor, income and expenditure. By applying anonymization methods available in the *sdcMicro* package, at least 20 different anonymization scenarios[7] were tested on each dataset. Most of the processing was done using a powerful server[8] and up to 16 – 20 processors (cores) at a time. Other processing platforms included a laptop and desktop computers, each using four processors. Computation times were significantly shorter for datasets processed on the server, compared to those processed on the laptop and desktop.

The use of parallelization can improve performance even on a single computer with one processor with multiple cores. Since *R* does not use multiple cores unless instructed to do so, our anonymization programs allowed for parallelization such that jobs/scenarios in each dataset could be processed simultaneously through efficient allocation of tasks to different processors. Without parallelization, depending on the server/computer, only one core is used when running the jobs sequentially. Running the anonymization program without parallelization leads to significantly longer execution time. Note however, that the parallelization itself also causes overhead. Therefore, a summation of the times it takes to run each task in parallel does not necessarily amount to the time it may take to run them sequentially. The fact that the RAM is shared might, however, slightly reduce the gains of parallelization. If you want to compare the results of different methods on large datasets that require long computation times, using parallel computing can be a solution.[9]

Appendix D zooms in on seven selected datasets from a health survey that were processed using the same parallelization program and anonymization methods. Note that the computation times in the appendix are only meant to create awareness for expected computation time, and may vary based on the type of computer used. In our case, although all datasets were anonymized using the parallelization program, computation times were significantly shorter for datasets processed on the server, compared to those processed on the laptop and desktop. Among those datasets processed on

[7] Here a scenario refers to a combination of SDC methods and their parameters.

[8] The server has 512 GB RAM and four processors each with 16 cores, translating to 64 cores total.

[9] The following website provides an overview of parallelization packages and solutions in *R*: http://cran.r-project.org/web/views/HighPerformanceComputing.html.

Note: Solutions are platform-dependent and therefore our solution is not further presented.

the server using the same number of processors (datasets 1, 2 and 6), some variation also exists in the computation times.

Note: Computation time in the table in Appendix D includes recalculating the risk after applying the anonymization methods, which is automatically done in sdcMicro when using standard methods/functions.

Using the function groupVars(), for instance, is not computationally intensive but can still take a long time if the dataset is large and risk measures have to be recalculated.

8.9 Common errors

In this section, we present a few common errors and their causes, which might be encountered when using the *sdcMicro* package in *R* for anonymization of microdata:

- The class of a certain variable is not accepted by the function, e.g., a categorical variable of class numeric should be first recoded to the required class (e.g., factor or data.frame). In the Section *Classes in R* is shown how to do this.
- After manually making changes to variables the risk did not change, since it is not updated automatically and has to be manually recomputed by using the function calcRisks().

CHAPTER

NINE

THE SDC PROCESS

This section presents the SDC process in a step-by-step representation and can be used as guidance for the actual SDC process. It should be noted, however, that jumping between steps and returning to previous steps is often required during the actual SDC process, as it is not necessarily a linear step-by-step process. This guidance brings together the different parts of the SDC process as discussed in the previous sections and links to these sections. The case studies in the next section follow these steps. This presentation is adapted from *HDFG12*. Fig. 9.1 at the end of this section presents the entire process in a schematic way.

9.1 Step 1: Need for confidentiality protection

Before starting the SDC process for a microdata set, the need for confidentiality protection has to be determined. This is closely linked to the interpretation of laws and regulations on this topic from the country in which the data originates and thus country-specific. A first step is to determine the statistical units in the dataset: if these are individuals, households or legal entities, such as companies, a need for disclosure control is likely. There are also examples of microdata for which there is no need for disclosure control. Examples could be data with climate and weather observations or data with houses as statistical units. Even if the primary statistical units are not natural or legal persons, however, the data can still contain confidential information on natural or legal persons. For example, a dataset with houses as primary statistical units can also contain information on the persons living in these houses and their income or a dataset on hospitalizations can include information about the hospitalized patients. In these cases, there is likely still a need for confidentiality protection. One option to solve this is to remove the information on the natural and legal persons in the datasets for release.

One dataset can also contain more than one type of statistical unit. The standard example here is a dataset containing both information on individuals and households. Another example is data with employees in enterprises. All types of statistical units present in the dataset have to be considered for the need of SDC. This is especially important in case the data has a hierarchical structure, such as individuals in households or employees in enterprises.

In addition, one has to evaluate whether the variables contained in the dataset are confidential or sensitive. Which variables are sensitive or confidential depends again on the applicable legislation and can differ substantially from country to country. In case the dataset includes sensitive or confidential variables, SDC is likely required. The set of sensitive variables and confidential variables together with the statistical units in the dataset determine the need for statistical disclosure control.

9.2 Step 2: Data preparation and exploring data characteristics

After assessing the need for statistical disclosure control, we should prepare the data and, if there are multiple, combine and consider all related data files. Then we explore the characteristics and structure in the data, which are important for the users of the data. Compiling an inventory of these characteristics is important for assessing the utility of the data after anonymization and producing an anonymized dataset, which is useful for end users.

The first step in data preparation is classifying the variables as sensitive or non-sensitive, and removing direct identifiers such as full names, passport numbers, addresses, phone numbers and GPS coordinates. In case of survey data, an inspection of the survey questionnaire is useful to classify the variables. Furthermore, it is necessary to select the variables that contain relevant information for end users and should be included in the dataset for release. At this point, it can also be useful to remove variables other than direct identifiers from the microdata set to be released. An example can be a variable with many missing values, e.g., a variable recorded only for a select group of individuals eligible for a particular survey module, and missing values for the rest. Such variables can cause a high level of disclosure risk while adding little information for end users. Examples are variables relating to education (current grade), where a missing value indicates that the individual is not currently in school, or variables relating to childbirth, where a missing value indicates that the individual has not delivered a child in the reference period. Missing values in themselves can be disclosive, especially if they indicate that the variable is not applicable. Often variables with the majority of values missing are deleted at this stage already. Other variables that might be deleted at this stage are those too sensitive to be anonymized and released or those not important to data users and that could increase the risk of disclosure.

Relationships may exist among variables in a dataset for a variety of reasons. For instance, variables can be mutually exclusive in cases where several binary variables are used for each category. An individual not in the labor force will have a missing value for the sector in which this person is employed (or more precisely not applicable). Relationships may also exist if some variables are ratios, sums or other mathematical functions of other variables. Examples are the variable household size (as a count of individuals per household), or aggregate expenditure (as a sum of all expenditure components). A certain value in one variable may also reduce the number of possible or valid values for another variable; for example, the age of an individual attending primary education or the gender of an individual having delivered a child. These relationships are important for two reasons: 1) they can be used by intruders to reconstruct anonymized values. For example, if age is suppressed but another variable indicates that they are in school, then it is still possible to infer a likely age range for that individual. Another example is if an individual is shown to be active in Sector B of the economy. Even if the labor status of this individual is suppressed, it can be inferred that this person is employed. 2) the relationships in the original data should be maintained in the anonymized dataset and inconsistencies should be avoided (e.g., SDC methods should not create 58-year-old school boys, or married 3-year-olds), or the dataset will be invalid for analysis. Another example is the case of expenditures per category, where it is important that the sum of the categories adds up to the total. One way to guarantee this is to anonymize the totals and then recalculate the sub-categories according to the original shares of the anonymized totals.

It is also useful at this stage to consolidate variables that provide the same information where possible, so as reduce the number of variables, reduce the likelihood of inconsistencies and minimize the variables an intruder can use to reconstruct the data. This is especially true if the microdata stems from an elaborate questionnaire and each variable represents one (sub-) question leading to a dataset with hundreds of variables. As an example, we take a survey with several labor force variables indicating whether a person is in the labor force, employed or unemployed, and if employed, in what sector. The data in Table 9.1 illustrates this example. It is possible that each type of sector has its own binary variable. In that case, this set of variables can be summarized in two variables: one variable indicating whether a person is in labor force and another indicating the employment status, as well as the respective sector if a person is employed. These two variables contain all information contained in the previous five variables and make the anonymization process easier. If data users are used to a certain release format where including all five variables has been the norm, then it is possible to transform the variables back after the anonymization process rather than complicating the anonymization process by trying to treat more variables than is necessary. This approach also guarantees that the relationships between the variables are preserved (e.g., no individuals will be employed in several sectors).

Table 9.1: Illustration of merging variables without information loss for SDC process

Before					After	
In labor force	Employed	Sector A	Sector B	Sector C	In labor force	Employed
Yes	Yes	Missing	Yes	Missing	Yes	B
No	No	Missing	Missing	Missing	No	No
Yes	Yes	Yes	Missing	Missing	Yes	A
Yes	Yes	Missing	Yes	Missing	Yes	B
Yes	Yes	Missing	Missing	Yes	Yes	C
Yes	No	Missing	Missing	Missing	Yes	No

Besides relationships between variables, we also gather information about the survey methodology, such as strata, sampling methods, survey design and sample weights. This information is important in later stages, when estimating the disclosure risk and choosing the SDC methods. It is important to distinguish between a full census and a sample. For a full census, it is common practice to publish only a sample, as the risk of disclosure for a full sample is too high, given that we know that everyone in the country or institution is in the data (see also the Section **'Special case: census data <anon_methods.html#Special case: census data'__**). Strata and sample weights can disclose information about the area or group to which an individual belongs (e.g., the weights can be linked with the geographical area or specific group in case of stratified sampling); this should be taken into account in the SDC process and checked before release of the dataset.

9.3 Step 3: Type of release

The type of release is an important factor for determining the required level of anonymization as well as the requirements end users have for the data (e.g., researchers need more detail than students for whom a teaching file might be sufficient) and should be clarified before the start of the anonymization process. Data release or dissemination by statistical agencies and data producers is often guided by the applicable law and dissemination strategies of the statistical agency, which specify the type of data that should be disseminated as well as the fashion.

Generally, there exist three types of data release methods for different target groups (the Section Release types provides more information on different release types):

- PUF: The data is directly available to anyone interested, e.g., on the website of the statistical agency
- SUF: The data is available to accredited researchers, who have to file their research proposals beforehand and have to sign a contract; this is also known as licensed file or microdata under contract
- Available in a controlled research data center: only on-site access to data on special computers; this is also known as data enclave

There are other data access possibilities besides these, such as teaching files or files for other specific purposes. Obviously, the required level of protection depends on the type of release: a PUF file must be protected to a greater extent than a SUF file, which in turn has to be protected more than a file which is available only in an on-site facility, since the options the intruder can use the data are limited in the latter case.

Besides the applicable legislation, the choice of the type of release depends on the type of the data and the content.

Note: Not every microdata set is suitable for release in any release type, even after SDC.

Some data cannot be protected sufficiently – it might always contain information that is too sensitive to be published as SUF or PUF. In such cases, the data can be released in on-site facilities, or the number of variables can be reduced by removing problematic variables.

Generally, the release of two or more anonymized datasets, e.g., tailored for different end users from the same original, is problematic because it can lead to disclosure if the two were later obtained and merged by the same user. The information contained in one dataset that is not contained in the other can lead to unintended disclosure. An exception is the simultaneous release and anonymization of a microdata set as PUF and SUF files. In this case, the PUF file is constructed from the SUF file by further anonymization. In that way, all information in the PUF file is also contained in the SUF file and the PUF file does not provide any additional information for users that have access to the SUF file.

Note: The anonymization process is an iterative process where steps can be revisited, whereas the publication of an anonymized dataset is a one-shot process.

Once the anonymized data is published, it is not possible to revoke and publish another dataset of the same microdata file. This would in fact mean publishing more than one anonymized file from the same microdata set, since some users might have saved the previous file.

9.4 Step 4: Intruder scenarios and choice of key variables

After determining the release type of the data, the possibilities of how an individual in the microdata could (realistically) be identified by an intruder under that release type should be examined. For PUF and SUF release the focus is on the use of external datasets from various source. These possibilities are described in disclosure or intruder scenarios, which specify what data an intruder could possibly have access to and how this auxiliary data can be used for identity disclosure. This leads to the specification of quasi-identifiers, which are a set of variables that are available both in the dataset to be released and in auxiliary datasets and need protection.

Note: If the number of quasi-identifiers is high, it is recommended to reduce the set of quasi-identifiers by removing some variables from the dataset for release.

This is especially true for PUF releases. Disclosure scenarios can also help define the required level of anonymization.

Drafting disclosure scenarios requires the support of subject matter specialists, assuming the subject specialist is not the same as the person doing the anonymization. Auxiliary datasets may contain information on the identity of the individuals and allow identity disclosure. Examples of such auxiliary data files are population registers and electoral rolls, as well as data collected by specialized firms.

Note: External datasets can come from many sources (e.g., other institutions, private companies) and it is sometimes difficult to make a full list of external data sources.

In addition, not all external data sources are in the public domain. Nevertheless, proprietary data can be used by the owner to re-identify individuals and should be taken into account in the SDC process, even if the exact content is not known. Also, the variables or datasets may not coincide perfectly (e.g., different definitions, more or less detailed variables, different survey period). Nevertheless, they should be considered in the SDC process.

Disclosure scenarios include both identity and inferential disclosure. The disclosure depends on the type of release, i.e., different data users have different data available and may use the data in a different way for re-identification. For example, a user in a research data center cannot match with large external datasets as (s)he is not permitted to take these into the data center. A user of a SUF is bound by an agreement specifying the use of the data and consequences if the agreement is breached (see the Section Release types). Furthermore, it should be evaluated whether, in case of a sample, possible intruders have knowledge as to which individuals are in the sample. This can be the case if it is known which schools were visited by the survey team, for example. A few examples of disclosure scenarios are (see the Section Disclosure scenarios for more information):

- Matching: The intruder uses auxiliary data, e.g., data on region, marital status and age from a population register, and matches them to released microdata. Individuals from the two datasets that match and are unique are successfully identified. This principle is used as an assumption in several disclosure risk measures, such as k-anonymity, individual and global risk, as described in the Section Measuring Risk. This scenario can apply to both PUFs and SUFs.
- Spontaneous recognition: This scenario should be considered for SUF files, but is especially important for data available in research data centers where outliers are present in the data and data is often not strongly anonymized. The researcher might (unintentionally) recognize some individuals he knows (e.g., his colleagues, neighbors, family members, public figures, famous persons or large companies), while inspecting the data. This is especially true for rare combinations of values, such as outliers or unlikely combinations.

9.5 Step 5: Data key uses and selection of utility measures

In this step, we analyze the main uses of the data by the end users of the released microdata file. The data should be useful for the type of statistical analysis for which the data was collected and for which it is mostly used. The uses and requirements of data users will be different for different release types. Contacting data users directly or searching for scientific studies and papers that use similar data can be useful when collecting this information and making this assessment. Alternatively, this information can be collected from research proposals by researchers when applying for microdata access (SUF) or user groups can be set up. Furthermore, it is important to understand what level of precision the data users need and what types of categories are used. For instance, in the case of global recoding of age in years, one could recode age in groups of 10 years, e.g., 0 – 9, 10 – 19, 20 – 29, ... Many indicators relating to the labor market use categories that span the range 15 – 65, however. Therefore, constructing categories that coincide with the categories used for the indicators keeps the data much more useful while at the same time reducing the risk of disclosure in a similar way. This knowledge is important for the selection of useful utility measures, which in turn are used for selecting appropriate SDC methods.

The uses of the data depend on the release type, too. Researchers using SUF files require a higher level of detail in the data than PUF users.

Note: Anonymization will always lead to information loss and a PUF file will have reduced utility. If certain users require a high level of detail, release types other than PUF should be considered, such as SUF or release through a research data center.

In the case of SUFs, it is easier to find the main uses of the data since access is documented. One way to obtain information on the use of PUF files is to ask for a short description of intended use of the data before supplying the data. This is, however, useful only if microdata has been released previously.

Statistics computed from the anonymized and released microdata file should produce analytical results that agree or almost agree with previously published statistics from the original data. If, for instance, a previously published primary school enrollment rate was computed from these data and published, the released anonymized dataset should produce a very similar result to the officially published result. At the very least, the result should fall within the confidence region of the published result. It might be the case that not all published statistics can be generated from the published data. If this is the case, a choice should be made on which indicators and statistics to focus, and inform the users as to which ones have been selected and why.

As discussed in the Section Measuring Utility and Information Loss, it is necessary to compute general utility measures that compare the raw and anonymized data, taking into consideration the end user's need for their analysis. In some cases the utility measures can give contradicting results, for example, a certain SDC method might lead to lower information loss for labor force figures but greater information loss for ratios relating to education. In such cases, the data uses might need to be ranked in order of importance and it should be clearly documented for the user that the prioritization of certain metrics over others means that certain metrics are no longer valid. This may be necessary, as

it is not possible to release multiple files for different users. This problem occurs especially in multi-purpose studies. For more details on utility measures, refer to the Section Measuring Utility and Information Loss.

Note on Steps 6 to 10

The following Steps 6 through 10 should be repeated if the data has quasi-identifiers that are on different hierarchical levels, e.g., individual and household. In that case, variables on the higher hierarchical level should be anonymized first, and then merged with the lower-level untreated variables. Subsequently, the merged dataset should be anonymized. This approach guarantees consistency in the treated data. If we neglect this procedure, the values of variables measured on the higher hierarchical level could be treated differently for observations of the same unit. For instance, the variable "region" is the same for all household members. If the value 'rural' would be suppressed for two members but not for the remaining three, this would lead to unintended disclosure; with the household ID the variable region would be easy to reconstruct for the two suppressed values. The Sections Household risk and Household structure provide more details on how to deal with data with household structure in *R* and *sdcMicro*.

9.6 Step 6: Assessing disclosure risk

The next step is to evaluate the disclosure risk of the raw data. Here it is important to distinguish between sample data and census data. In the case of census data, it is possible to directly calculate the risk measures when assuming that the dataset covers the entire population. If working with a sample, or a sample of the census (which is the more common case when releasing sample data), we can use the models discussed in the Section Measuring Risk to estimate the risk in the population. The main inputs for the risk measurement are the set of quasi-identifiers determined from the disclosure scenarios in Step 4 and the thresholds for risk calculations (e.g., the level of k-anonymity or the threshold for which an individual is considered at risk). If the data has a hierarchical structure (e.g., a household structure), the risk should be measured taking into account this structure as described the Section Household risk.

The different risk measures described in the Section Measuring Risk each have advantages and disadvantages. Generally, k-anonymity, individual risk and global risk are used to produce an idea of the disclosure risk. These values can initially be very high but can often very easily be reduced after some simple but appropriate recoding (see *Step 8: Choice and application of SDC methods*). The thresholds shall be determined according to the release type. Always remember, though, that when using a sample, the risk measures based on the models presented in the literature offer a worst-case risk scenario and might therefore be an exaggeration of the real risks for some cases (see the Section Individual risk).

9.7 Step 7: Assessing utility measures

To quantify the information loss due to the anonymization, we first compute the utility measures selected in Step 6 using the raw data. This creates a base for comparison of results obtained when using the anonymized data – i.e., in Step 10.

Note: If the raw data is a sample, the utility measures are an estimate with a variance and therefore it is useful to construct confidence intervals in addition to the point estimates for the utility measures.

9.8 Step 8: Choice and application of SDC methods

The choice of SDC methods depends on the need for data protection (as measured by the disclosure risk), the structure of the data and the type of variables. The influence of different methods on the characteristics of the data important

for the users or the data utility should also be taken into account when selecting the SDC methods. In practice, the choice of SDC methods is partially a trial-and-error process: after applying a chosen method, disclosure risk and data utility are measured and compared to other choices of methods and parameters. The choice of methods is bound by legislation on the one hand, and a trade-off between utility and risk on the other.

The classification of methods as presented in Table 6.1 gives a good overview for choosing the appropriate methods. Methods should be chosen according to the type of variable – continuous or categorical – the requirements by the users and the type of release. The anonymization of datasets with both continuous and categorical variables is discussed in the Section Classification of variables.

In general for anonymization of categorical variables, it is useful to restrict the number of suppressions by first applying global recoding and/or removing variables from the microdata set. When the required number of suppressions to achieve the required level of risk is sufficiently low, the few individuals at risk can be treated by suppression. These are generally outliers. It should be noted that possibly not all variables can be released and some must be removed from the dataset (see *Step 2: Data preparation and exploring data characteristics*). Recoding and minimal use of suppression ensures that already published figures from the raw data can be reproduced sufficiently well from the anonymized data. If suppression is applied without sufficient recoding, the number of suppressions can be very high and the structure of the data can change significantly. This is because suppression mainly affects combinations that are rare in the data.

If the results of recoding and suppression do not achieve the required result, especially in cases where the number of select quasi-identifiers is high, an alternative is using perturbative methods. These can be used without prior recoding of variables. These methods, however, preserve data structure only partially. The preferred method depends on the requirements of the users. We refer to the Section Anonymization Methods and especially the Section Perturbative methods for a discussion of perturbative methods implemented in *sdcMicro*.

Finally, the choice of SDC methods depends on the data used since the same methods produce different results on different datasets. Therefore, the comparison of results with respect to risk and utility (Steps 9 and 10) is key to the choice made. Most methods are implemented in the *sdcMicro* package. Nevertheless, it is sometimes useful to use custom-made solutions. A few examples are presented in the Section Anonymization Methods.

9.9 Step 9: Re-measure risk

In this step, we re-evaluate the disclosure risk with the risk measures chosen under Step 6 after applying SDC methods. Besides these risk measures, it is also important to look at individuals with high risk and/or special characteristics, combinations of values or outliers in the data. If the risk is not at an acceptable level, Steps 6 to 10 should be repeated with different methods and/or parameters.

Note: Risk measures based on frequency counts (k-anonymity, individual risk, global risk and household risk) cannot be used after applying perturbative methods since their risk estimates are not valid.

These methods are based on introducing uncertainty into the dataset and not on increasing the frequencies of keys in the data and will hence overestimate the risk.

9.10 Step 10: Re-measure utility

In this step, we re-measure the utility measures from Step 7 and compare these with the results from the raw data. Also, it is useful here to construct confidence intervals around the point estimates and compare these confidence intervals. The importance of the absolute value of a deviation can only be interpreted knowing the variance of the estimate. Besides these specific utility measures, general utility measures, as discussed in the Section Measuring Utility and Information Loss , should be evaluated. This is especially important if perturbative methods have been applied. If

the data does not meet the user requirements and deviations are too large, repeat Steps 6 to 10 with different methods and/or different parameters.

Note: Anonymization will always lead to at least some information loss.

9.11 Step 11: Audit and Reporting

After anonymization, it is important to check whether all relationships in the data as identified in Step 2, such as variables that are sums of other variables or ratios, are preserved. Also, any unusual values caused by the anonymization should be detected. Examples of such anomalies are negative income or a pupil in the twentieth grade of school. This can happen after applying perturbative SDC methods. Furthermore, it is necessary to check whether previously published indicators from the raw data are reproducible from the data to be released. If this is not the case, data users might question the credibility of the anonymized dataset.

An important step in the SDC process is reporting, both internal and external. Internal reporting contains the exact description of anonymization methods used, parameters but also the risk measures before and after anonymization. This allows replication of the anonymized dataset and is important for supervisory authorities/bodies to ensure the anonymization process is sufficient to guarantee anonymity according to the applicable legislation.

External reporting informs the user that the data has been anonymized, provides information for valid analysis on the data and explains the limitations to the data as a result of the anonymization. A brief description of the methods used can be included. The release of anonymized microdata should be accompanied by the usual metadata of the survey (survey weight, strata, survey methodology) as well as information on the anonymization methods that allow researchers to do valid analysis (e.g., amount of noise added, transition matrix for PRAM).

Note: Care should be taken that this information cannot be used for re-identification (e.g., no release of random seed for PRAM).

The metadata must be updated to comply with the anonymized data. Variable descriptions or value labels might have changed as a result of the anonymization process. In addition, the information loss due to the anonymization process should be explained in detail to the users to make them aware of the limits to the validity of the data and their analyses.

9.12 Step 12: Data release

The last step in the SDC process is the actual release of the anonymized data. This step depends on the type of release chosen in Step 3. Changes to the variables made under Step 2, such a merging variables, can be undone to generate a dataset useful for users.

Recommended Reading Material on Risk Measurement

Dupriez, Olivier, and Ernie Boyko. 2010. *Dissemination of Microdata Files; Principles, Procedures and Practices.* IHSN Working Paper No. 005, International Household Survey Network (IHSN). http://www.ihsn.org/HOME/sites/default/files/resources/IHSN-WP005.pdf

References

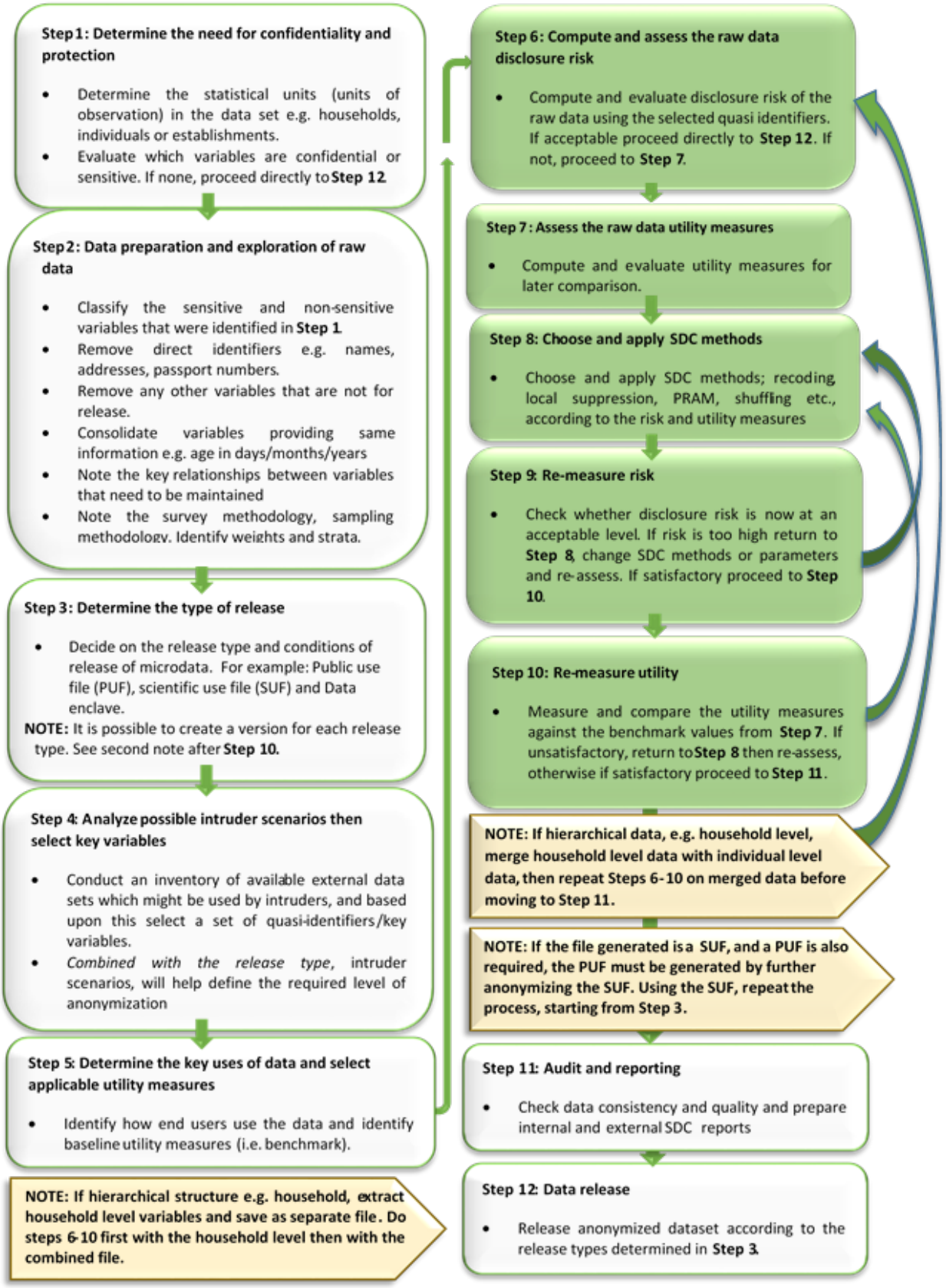

Fig. 9.1: Overview of the SDC process

CHAPTER

TEN

APPENDICES

10.1 Appendix A: Overview of Case Study Variables

.	Variable	Description
1	REGION	Region
2	DIST	District
3	URBRUR	Area of residence
4	WGTHH	Individual weighting coefficient (country-specific weighting co-efficient to derive individual-level indi
5	WGTPOP	Population weighting coefficient (weighting co-efficient to derive population-level indicators)
6	IDH	Household unique identification
7	IDP	Individual identification
8	HHSIZE	Household members
9	GENDER	Sex
10	REL	Relationship to household head
11	MARITAL	Marital status
12	AGEYRS	Age in completed years
13	AGEMTH	Age of child in completed months
14	RELIG	Religion of household head
15	ETHNICITY	Ethnicity
16	LANGUAGE	Language
17	MORBID	Morbidity last x weeks
18	MEASLES	Child immunized against Measles
19	MEDATT	Sought medical attention
20	CHWEIGHTKG	Weight of the child (Kg)
21	CHHEIGHTCM	Height of the child (cms)
22	ATSCHOOL	Current school enrolment
23	EDUCY	Highest level of education completed
24	EDYRS	Years of education
25	EDYRSCURRAT	Years of education for currently enrolled
26	SCHTYP	Type of school attending
27	LITERACY	Literacy status
28	EMPTYP1	Type of employment, Primary job
29	UNEMP1	Unemployed
30	INDUSTRY1	1 digit industry classification, Primary job
31	EMPCAT1	Employment categories, Primary job
32	WHOURSWEEK1	Hours worked last week, Primary job
33	OWNHOUSE	Ownership of dwelling unit
34	ROOF	Main material used for roof

Continued

Table 10.1 – continued from previous page

.	Variable	Description
35	TOILET	Main toilet facility
36	ELECTCON	Connection of electricity in dwelling
37	FUELCOOK	Main cooking fuel
38	WATER	Main source of water
39	OWNAGLAND	Ownership of agricultural land
40	LANDSIZEHA	Land size owned by household (ha)
41	OWNMOTORCYCLE	Ownership of motorcycle
42	CAR	Ownership of car
43	TV	Ownership of television
44	LIVESTOCK	Number of large-sized livestock owned
45	INCRMT	Total amount of remittances received from remittance sending members
46	INCWAGE	Wage and salaries (annual)
47	INCBONSOCALL	Bonus and social allowance from wage job (annual)
48	INCFARMBSN	Gross income from household farm businesses (annual)
49	INCNFARMBSN	Gross income from household non-farm businesses (annual)
50	INCRENT	Rental income (annual)
51	INCFIN	Financial income from savings, loans, tax refunds, maturity payments on insurance
52	INCPENSN	Pension and other social assistance (annual)
53	INCOTHER	Other income(annual)
54	INCTOTGROSSHH	Total gross household income (annual)
55	FARMEMP	Farm employment
56	THOUSEXP	Total expenditure on housing
57	TFOODEXP	Total food expenditure
58	TALCHEXP	Total alcohol expenditure
59	TCLTHEXP	Total expenditure on clothing and footwear
60	TFURNEXP	Total expenditure on furnishing
61	THLTHEXP	Total expenditure on health
62	TTRANSEXP	Total expenditure on transport
63	TCOMMEXP	Total expenditure on communications
64	TRECEXP	Total expenditure on recreation
65	TEDUEXP	Total expenditure on education
66	TRESTHOTEXP	Total expenditure on restaurants and hotel
67	TMISCEXP	Total miscellaneous expenditure
68	TANHHEXP	Total annual nominal household expenditures

10.2 Appendix B: Example of Blanket Agreement for SUF

Source: *DuBo10*

10.3 Appendix C: Internal and External Reports for Case Studies

This appendix provides example for internal and external reports on the anonymization process for the case studies in the Section Case studies. The internal report consists of two parts: the first is on the anonymization of the household-level variables and the second is on the anonymization of the individual-level variables.

Agreement between [providing agency] and [receiving agency] regarding the deposit and use of microdata

A. This agreement relates to the following microdatasets:
1. _____
2. _____
3. _____
4. _____
5. _____

B. Terms of the agreement:

As the owner of the copyright in the materials listed in section A, or as duly authorized by the owner of the copyright in the materials, the representative of [providing agency] grants the [receiving agency] permission for the datasets listed in section A to be used by [receiving agency] employees, subject to the following conditions:

1. Microdata (including subsets of the datasets) and copyrighted materials provided by the [providing agency] will not be redistributed or sold to other individuals, institutions or organisations without the [providing agency]'s written agreement. Non-copyrighted materials which do not contain microdata (such as survey questionnaires, manuals, codebooks, or data dictionaries) may be distributed without further authorization. The ownership of all materials provided by the [providing agency] remains with the [providing agency].
2. Data will be used for statistical and scientific research purposes only. They will be employed solely for reporting aggregated information, including modeling, and not for investigating specific individuals or organisations.
3. No attempt will be made to re-identify respondents, and there will be no use of the identity of any person or establishment discovered inadvertently. Any such discovery will be reported immediately to the [providing agency].
4. No attempt will be made to produce links between datasets provided by the [providing agency] or between [providing agency] data and other datasets that could identify individuals or organisations.
5. Any books, articles, conference papers, theses, dissertations, reports or other publications employing data obtained from the [providing agency] will cite the source, in line with the citation requirement provided with the dataset.
6. An electronic copy of all publications based on the requested data will be sent to the [providing agency].
7. The [providing agency] and the relevant funding agencies bear no responsibility the data's use or for interpretation or inferences based upon it.
8. An electronic copy of all publications based on the requested data will be sent to the [providing agency].
9. Data will be stored in a secure environment, with adequate access restrictions. The [providing agency] may at any time request information on the storage and dissemination facilities in place.
10. The [recipient agency] will provide an annual report on uses and users of the listed microdatasets to the [providing agency], with information on the number of researchers having accessed each dataset, and on the output of this research.
11. This access is granted for a period of [provide information on this period, or state that the agreement is open ended].

C. Communications:
The [receiving organisation] will appoint a contact person who will act as unique focal person for this agreement. Should the focal person be replaced, the [recipient agency] will immediately communicate the name and coordinates of the new contact person to the [providing agency]. Communications for administrative and procedural purposes may be made by email, fax or letter as follows:

Communications made by [providing agency] to [recipient agency] will be directed to:
Name of contact person:
Title of contact person:
Address of the recipient agency:
Email:
Tel:
Fax:

Communications made by [recipient agency] to [depositor agency]
will be directed to:

Name of contact person:
Title of contact person:
Address of the recipient agency:
Email:
Tel:
Fax:

D. Signatories
The following signatories have read and agree with the Agreement as presented above:

Representative of the [providing agency]
Name _____
Signature _____ Date _____

Representative of the [recipient agency]
Name _____
Signature _____ Date _____

10.3.1 Case study 1 - Internal report

SDC report (adapted from the report function in sdcMicro)

The dataset consists of 10,574 observations (i.e., 10,574 individuals in 2,000 households).

Household-level variables

Anonymization methods applied to household-level variables:

- Removing households of size larger than 13 (29 households)
- Local suppression to achieve 2-anonymity, with importance vector to prevent suppressing values of the variables HHSIZE, REGION and URBRUR
- Recoding the variable LANDSIZEHA: rounding to one digit for values smaller than 1, rounding to zero digits for other values, grouping values 5-19 and 20-40, topcoding at 40
- PRAMming the variables ROOF, TOILET, WATER, ELECTCON, FUELCOOK, OWNMOTORCYCLE, CAR, TV and LIVESTOCK
- Noise addition (level 0.01 and 0.05 for outliers) to the income and expenditure components, replacing aggregates by sum of perturbed components

Selected (key) variables:

Household level					
categorical	URBRUR	REGION	HHSIZE	OWNAGLAND	RELIG
continuous	LAND-SIZEHA	TANHH-EXP	TFOODEXP	TALCHEXP	TCLTHEXP
.	THOUSEXP	TFURN-EXP	THLTHEXP	TTRANSEXP	TCOMMEXP
.	TRECEXP	TEDUEXP	TRESHOT-EXP	TMISCEXP	INCTOT-GROSSHH
.	INCRMT	INCWAGE	INC-FARMBSN	INCN-FARMBSN	INCRENT
.	INCFIN	INCPENSN	INCOTHER		
weight	WGTPOP				
hhID	not defined				
strata	not defined				

- Modifications on categorical key variables: TRUE
- Modifications on continuous key variables: TRUE
- Modifications using PRAM: TRUE
- Local suppressions: TRUE

Disclosure risk (household-level variables):

Frequency analysis for categorical key variables: 181

Number of observations violating 2-Anonymity: 0 (unmodified data: 103)

3-Anonymity: 104 (unmodified data: 229)

5-Anonymity: 374 (unmodified data: 489)

Percentage of observations violating

2-Anonymity: 0% (unmodified data: 5.15%)

3-Anonymity: 5.28% (unmodified data: 11.45%)

5-Anonymity: 18.7% (unmodified data: 24.45%)

Disclosure risk categorical variables:

Expected Percentage of Re-identifications: 0.05161614% (~ 1.0 observations)

(unmodified data: 0.001820465% (~ 0.36 observations))

10 combinations of categories with highest risk:

.	URBRUR	REGION	HHSIZE	OWNAGLAND	RELIG	fk	Fk
1	2	6	2	3	7	1	372.37
2	1	5	1	1	6	1	226.35
3	2	5	2	3	6	1	430.21
4	2	2	1	1	NA	1	173.05
5	2	6	1	1	5	1	80.05
6	1	6	1	3	5	1	343.27
7	2	5	1	2	NA	1	140.60
8	2	6	1	3	7	1	230.29
9	2	5	12	1	9	1	475.01
10	2	6	3	1	1	1	338.57

10.3.2 Case study 1 - External report

This case study microdata set has been treated to protect confidentiality. Several methods have been applied to protect the confidentiality: removing variables from the original dataset, removing records from the dataset, reducing detail in variables by recoding and top-coding, removing particular values of individuals at risk (local suppression) and perturbing values of certain variables.

Removing variables

The released microdata set has only a selected number of variables contained in the initial survey. Not all variables could be released in this SUF without breaching confidentiality rules.

Removing records

To protect confidentiality, records of households larger than 13 were removed. Thirty households out of a total of 2,000 households in the dataset were removed. Reducing detail in variables by recoding and top-coding The variable LANDSIZEHA was rounded to one digit for values smaller than 1, rounded to zero digits for other values, grouped for values 5-19 and 20-40 and topcoded at 40. The variable AGEYRS was recoded to ten-year age intervals for values in the age range 15 – 65.

Local suppression

Values of certain variables for particular households and individuals were deleted. In total, six values of the variable URBRUR, one of the REGION variable, 48 for the OWNAGLAND variable, 16 for the RELIG variable and 323 values of the variable REL were deleted.

Perturbing values

Uncertainty was introduced in the variables ROOF, TOILET, WATER, ELECTCON, FUELCOOK, OWNMOTORCYCLE, CAR, TV and LIVESTOCK by using the PRAM method. This method changes a certain percentage of values of variables within each variable. Here invariant PRAM was used, which guarantees that the univariate tabulations stay unchanged. Multivariate tabulations may be changed. Unfortunately, the transition matrix cannot be published. The income and expenditure variables were perturbed by adding noise (adding small random values to the original values). The noise added was 0.01 times the standard deviation in the original data and 0.05 for outliers. Noise was added to

the components and the aggregates were recomputed to guarantee that the proportions of the different components did not change.

10.3.3 Case study 2 - Internal report

SDC report (adapted from the report function in sdcMicro)

This report describes the anonymization measures for the PUF release additional to those already taken in the first case study. Therefore, this report should be read in conjunction with the internal report for case study 1. The original dataset consists of 10,574 observations (i.e., 10,574 individuals in 2,000 households). The dataset used for the anonymization of the PUF file is the anonymized SUF file from case study 1. This dataset consists of 10.068 observations in 1,970 households. The difference is due to the removal of large households and sensitive or identifying variables in the first case study.

Household-level variables

Anonymization methods applied to household-level variables:

- **For SUF release (see case study 1):**
  - Removing households of size larger than 13 (29 households)
  - Local suppression to achieve 2-anonymity, with importance vector to prevent suppressing values of the variables HHSIZE, REGION and URBRUR
- **For PUF release:**
  - Remove variables OWNLANDAG, RELIG and LANDSIZEHA
  - Local suppression to achieve 5-anonymity, with importance vector to prevent suppressing values of the variables HHSIZE and REGION
  - PRAMming the variables ROOF, TOILET, WATER, ELECTCON, FUELCOOK, OWNMOTORCYCLE, CAR, TV and LIVESTOCK
  - Create deciles for aggregate income and expenditure (TANNEXP and INCTOTGROSSHH) and replace the actual values with the mean of the corresponding decile. Replace income and expenditure components with the proportion of original totals.

Selected (key) variables:

10.3.4 Case study 2 - External report

This case study microdata set has been treated to protect confidentiality. Several methods have been applied to protect the confidentiality: removing variables from the original dataset, removing records from the dataset, reducing detail in variables by recoding and top-coding, removing particular values of individuals at risk (local suppression) and perturbing values of certain variables.

Removing variables

The released microdata set has only a selected number of variables contained in the initial survey. Not all variables could be released in this PUF without breaching confidentiality rules.

Removing records

To protect confidentiality, records of households larger than 13 were removed. Twenty-nine households out of a total of 2,000 households in the dataset were removed.

Reducing detail in variables by recoding and top-coding

The variable AGEYRS was recoded to ten-year age intervals for values in the age range 15 – 65 and bottom- and top-coded at 15 and 65. The variables REL, MARITAL, EDUCY and INDUSTRY1 were recoded to less detailed categories. The total income and expenditure variables were recoded to the mean of the corresponding deciles and the income and expenditure components to the proportion of the totals.

Local suppression

Values of certain variables for particular households and individuals were deleted. In total, 67 values of the variable URBRUR, 126 of the REGION variable, 91 for the AGEYRS variable and 323 values of the variable REL were deleted.

Perturbing values

Uncertainty was introduced in the variables ROOF, TOILET, WATER, ELECTCON, FUELCOOK, OWNMOTORCYCLE, CAR, TV and LIVESTOCK by using the PRAM method. This method changes a certain percentage of values of variables within each variable. Here invariant PRAM was used, which guarantees that the univariate tabulations stay unchanged. Multivariate tabulations may be changed. Unfortunately, the transition matrix cannot be published.

10.4 Appendix D: Execution Times for Multiple Scenarios Tested using Selected Sample Data

References

Datasets	1	2	3	4	5	6
Total Observations	40,449	102,429	20,261	34,785	23,022	26,411
Quasi-identifiers	Number of categories*					
RURAL/URBAN	2	2	2	2	2	2
REGION	8	2	10	5	4	10
HOUSEHOLD SIZE	25	26	18	17	20	18
OWNHOUSE	2	NA	2	2	3	2
SEX	2	2	2	2	2	2
RELATION TO HEAD OF HOUSEHOLD	8	7	8	7	9	7
MARITAL STATUS	5	6	6	6	6	6
AGE IN COMPLETED YEARS	99	98	97	98	109	99
ETHNICITY	NA	NA	NA	NA	10	20
RELIGION	NA	NA	NA	4	10	11
HIGHEST LEVEL OF EDUC COMPLETED	7	7	9	6	8	8
YEARS OF EDUCATION - ENROLLED	16	17	17	18	14	NA
EMPLOYMENT TYPE	7	7	6	5	6	7
LITERACY	2	NA	2	3	2	2
INDUSTRY CODE	10	464	10	11	10	10
CURRENTLY ENROLLED AT SCHOOL	2	2	2	2	2	2
Processing Platform	Server: 16 Processors	Desktop: 2 Processors	Desktop : 2 Processors	Desktop: 2 Processors	Server: 16 Processors	Laptop: 4 Processors
Total execution time with parallelization (hours)	6.11	10.75	23.70	67.58	77.14	282.58
Scenarios						
base	0	0	0	0	0	0
1	0.35	0.72	0.15	0.31	0.36	0.30
2	0.37	0.75	0.16	0.32	0.37	0.31
3	0.38	0.78	0.16	0.33	0.37	0.31
4	0.39	0.81	0.16	0.33	0.38	0.32
5	4.14	6.46	2.27	6.07	34.85	163.59
6	1.34	1.37	1.02	2.36	22.58	131.91
7	0.82	0.51	0.96	2.34	16.28	70.63
8	0.88	0.63	0.82	2.17	19.88	130.50
9	2.99	2.75	1.77	5.01	43.21	177.47
10	1.51	0.80	1.23	4.06	17.99	84.16
11	1.52	0.61	1.11	4.19	24.39	114.03
12	4.03	6.45	2.25	5.94	31.38	163.42
13	1.36	1.34	0.99	2.33	23.47	129.24
14	0.78	0.54	0.96	2.31	15.83	74.24
15	0.90	0.67	0.83	2.13	21.59	123.70
16	2.82	2.84	1.74	5.00	40.50	179.26
17	1.61	0.79	1.21	4.04	19.00	82.80
18	1.48	0.62	1.11	4.31	24.11	118.77
19	4.06	6.51	2.21	6.07	31.79	159.04
20	1.25	1.43	0.96	2.35	21.36	130.83
21	0.78	0.54	0.94	2.36	16.44	69.55
22	0.91	0.68	0.80	2.18	22.74	131.49
23	2.82	2.86	1.73	4.98	42.32	168.45
24	1.49	0.81	1.20	4.04	19.37	84.13
25	1.40	0.64	1.08	4.26	24.54	113.05
26	3.94	6.36	2.19	5.99	31.00	145.07
27	1.36	1.43	0.96	2.31	23.18	107.94
28	0.79	0.54	0.93	2.30	16.96	73.40
29	0.90	0.70	0.81	2.13	22.25	118.46
30	2.86	2.80	1.79	5.05	40.63	137.30
31	1.45	0.80	1.24	4.11	19.01	80.12
32	1.60	0.65	1.14	4.39	23.87	95.11
33	3.92	6.40	2.31	6.13	29.99	110.22
34	1.27	1.43	1.02	2.35	22.61	97.67
35	0.80	0.54	0.99	2.32	16.32	72.85
36	0.89	0.70	0.85	2.14	19.68	101.54
37	2.57	2.82	1.92	4.99	37.70	104.72
38	1.34	0.80	1.35	4.03	18.64	70.92
39	1.42	0.65	1.17	4.17	23.10	70.46
Approximated execution time if sequentially processed (hours)	65.48	69.52	46.46	132.20	880.06	3987.25

*NA indicates that the variable is not available in that dataset and, hence, could not be selected as quasi-identifier.

Description of Scenarios				
Scenario	Recode	Local suppression	Expenditure variables	PRAM
base	-	-	-	-
1	-	k=3	-	PRAM on roof, water, toilet
2	-	k=3	add noise to expenditure variables	PRAM on roof, water, toilet
3	-	k=3	shuffle expenditure variables	PRAM on roof, water, toilet
4	topcode age	k=3	add noise to aggregate expenditures	-
5	recode age	k=3	-	-
6	recode age	k=3	-	-
7	recode age	k=3, importance age	-	-
8	recode age	k=3, importance ethnicity and religion	-	-
9	recode age	k=5	-	-
10	recode age	k=5, importance age	-	-
11	recode age	k=5, importance ethnicity and religion	-	-
12	topcode age	k=3	-	PRAM on roof, water, toilet
13	topcode age	k=3	-	PRAM on roof, water, toilet
14	recode age	k=3, importance age	-	PRAM on roof, water, toilet
15	recode age	k=3, importance ethnicity and religion	-	PRAM on roof, water, toilet
16	recode age	k=5	-	PRAM on roof, water, toilet
17	recode age	k=5, importance age	-	PRAM on roof, water, toilet
18	recode age	k=5, importance ethnicity and religion	-	PRAM on roof, water, toilet
19	recode age	k=3	add noise to expenditure variables	PRAM on roof, water, toilet
20	recode age	k=3	add noise to expenditure variables	PRAM on roof, water, toilet
21	recode age	k=3, importance age	add noise to expenditure variables	PRAM on roof, water, toilet
22	recode age	k=3, importance ethnicity and religion	add noise to expenditure variables	PRAM on roof, water, toilet
23	recode age	k=5	add noise to expenditure variables	PRAM on roof, water, toilet
24	recode age	k=5, importance age	add noise to expenditure variables	PRAM on roof, water, toilet
25	recode age	k=5, importance ethnicity and religion	add noise to expenditure variables	PRAM on roof, water, toilet
26	topcode age	k=3	shuffle expenditure variables	PRAM on roof, water, toilet
27	recode age	k=3	shuffle expenditure variables	PRAM on roof, water, toilet
28	recode age	k=3, importance age	shuffle expenditure variables	PRAM on roof, water, toilet
29	recode age	k=3, importance ethnicity and religion	shuffle expenditure variables	PRAM on roof, water, toilet
30	recode age	k=5	shuffle expenditure variables	PRAM on roof, water, toilet
31	recode age	k=5, importance age	shuffle expenditure variables	PRAM on roof, water, toilet
32	recode age	k=5, importance ethnicity and religion	shuffle expenditure variables	PRAM on roof, water, toilet
33	topcode age	k=3	add noise to aggregate expenditures	PRAM on roof, water, toilet
34	recode age	k=3	add noise to aggregate expenditures	PRAM on roof, water, toilet
35	recode age	k=3, importance age	add noise to aggregate expenditures	PRAM on roof, water, toilet
36	recode age	k=3, importance ethnicity and religion	add noise to aggregate expenditures	PRAM on roof, water, toilet
37	recode age	k=5	add noise to aggregate expenditures	PRAM on roof, water, toilet
38	recode age	k=5, importance age	add noise to aggregate expenditures	PRAM on roof, water, toilet
39	recode age	k=5, importance ethnicity and religion	add noise to aggregate expenditures	PRAM on roof, water, toilet

Fig. 10.1: Description of anonymization scenarios

CHAPTER

ELEVEN

ACKNOWLEDGEMENTS

Authors of this guide:

- Thijs Benschop, Consultant The World Bank
- Cathrine Machingauta, The World Bank
- Matthew Welch, The World Bank

Correspondence: Matthew Welch, mwelch@worldbank.org

Acknowledgments: The authors thank Olivier Dupriez (The World Bank) for his technical comments and inputs throughout the process.

The production of this guide was made possible through a World Bank Knowledge for Change II Grant: KCP II - A microdata dissemination challenge: Balancing data protection and data utility. Grant number: TF 015043, Project Number P094376. As well as from United Kingdom - DFID funding to the World Bank Multi-Donor Trust Fund - International Household Survey and Accelerated Data Program – TF071804/TF011722.

CHAPTER

TWELVE

CASE STUDIES (ILLUSTRATING THE SDC PROCESS)

In order to evaluate the use of different SDC methods on different types of survey datasets, we compared the results of the different methods applied to 75 datasets from 52 countries representing six geographic regions: Latin America and the Caribbean (LAC), Sub-Saharan Africa (AFR), South Asia (SA), Europe and Central Asia (ECA), Middle East and North Africa (MENA) as well as East Asia and the Pacific (EAP). The datasets chosen were from a mix of datasets that are already publically available, as well as data made available only to the World Bank for official business. The surveys used included, amongst others, household, demographic, and health surveys. The variables from these surveys used for the experiments were selected based on their relevance for users (e.g., for indicators, MDGs), their sensitivity, and their classification with respect to the SDC process.

The following case studies draw from knowledge gained from these experiments and try to incorporate the lessons learned. The case studies use synthetic data that mimic the structure of the survey types we used in our experiments and present the anonymization of a dataset similar to many surveys designed to measure household income and consumption, labor force participation and general demographic characteristics. The first case study creates a SUF, whereas in the second case study we take this SUF and treat it further to create a PUF.

12.1 Case study 1- SUF

This case study shows an example of how the anonymization process might be approached, particularly for a dataset with many continuous variables. We also show how this can be achieved using the open source and free *sdcMicro* package and *R*. A ready-to-run *R* script for this case study and the dataset are also available to reproduce the results and allow the user to adapt the code (see http://ihsn.org/home/projects/sdc-practice). Extracts of this code are presented in this section to illustrate several steps of the anonymization process.

Note: The choices of methods and parameters in this case study are based on this particular dataset and the results and choices might be different for other datasets.

The aim is to show the process, not to compare methods per se.

This example uses a dataset with a similar structure to that of a typical social survey with a focus on demographics, labor force participation and income and expenditure patterns. The dataset has been compiled using observations from several datasets from different countries. They are considered synthetic data and as such are used only for illustrative purposes. The source datasets were already treated for disclosure control by their producers. This does not matter, as our concern is to illustrate the process only. The data from which we compiled our case study file was from surveys that contain many variables, but pay particular attention to labor force variables as well as household income and household expenditure variables. The variables in the demo dataset have already been pre-selected from the total set of variables available in the datasets. See Appendix A for the complete overview of all variables.

This case study follows the steps of the SDC process outlined in the Section The SDC Process.

12.1.1 Step 1: Need for disclosure control

The statistical units in this dataset are individuals and households. The household structure provides a hierarchical structure in the data, which should be taken into account when measuring risk and selecting anonymization methods.

The data contains variables with demographic information, income, expenditures, education variables and variables relating to the labor status of the individual. These variables include sensitive and confidential variables. The dataset is an example of a social survey and, due to the nature of the statistical units and the variables, disclosure control is needed before release of the microdata. This is the case regardless of the legal framework, which is not specified here, as this is a hypothetical dataset.

12.1.2 Step 2: Data preparation and exploring data characteristics

The first step is to explore the data. To analyze the data in *R* we first have to read the data into *R*. In our case, the data is saved in a *STATA* file (.dta file). To read *STATA* files, we need to load the *R* package *foreign* (see the Section Read functions in R on importing other data formats in *R*). We also load the *sdcMicro* package and several other packages used later for the computation of the utility measures. If these packages are not yet installed, you should do so before trying to load them. The *R* code for this case study demonstrates how to do this.

Listing 12.1: Loading required packages

```
# Load required packages
library(foreign)    # for read/write function for STATA files
library(sdcMicro)   # sdcMicro package with functions for the SDC process
library(laeken)     # for GINI
library(reldist)    # for GINI
library(bootstrap)  # for bootstrapping
library(ineq)       # for Lorenz curves
```

After setting the working directory to the directory where the *STATA* file is stored, we load the data into the object called *file*. All output, unless otherwise specified, is saved in the working directory.

Listing 12.2: Loading the data

```
#  Set working directory
setwd("C:/WorldBank/CaseStudy/")

# Specify file name
fname <- " case_1_data.dta"

# Read-in file
file <- read.dta(fname, convert.factors = F) # factors as numeric code
```

We check the number of variables, number of observations and variable names, as shown in Listing 12.3.

Listing 12.3: Number of individuals and variables and variable names

```
dim(file) # Dimensions of file (observations, variables)
## [1] 10574 68

colnames(file) # Variable names
##  [1] "REGION"        "DIST"          "URBRUR"        "WGTHH"
##  [5] "WGTPOP"        "IDH"           "IDP"           "HHSIZE"
##  [9] "GENDER"        "REL"           "MARITAL"       "AGEYRS"
## [13] "AGEMTH"        "RELIG"         "ETHNICITY"     "LANGUAGE"
## [17] "MORBID"        "MEASLES"       "MEDATT"        "CHWEIGHTKG"
```

(continues on next page)

(continued from previous page)

```
## [21] "CHHEIGHTCM"     "ATSCHOOL"       "EDUCY"          "EDYRS"
## [25] "EDYRSCURRAT"    "SCHTYP"         "LITERACY"       "EMPTYP1"
## [29] "UNEMP1"         "INDUSTRY1"      "EMPCAT1"        "WHOURSWEEK1"
## [33] "OWNHOUSE"       "ROOF"           "TOILET"         "ELECTCON"
## [37] "FUELCOOK"       "WATER"          "OWNAGLAND"      "LANDSIZEHA"
## [41] "OWNMOTORCYCLE" "CAR"            "TV"             "LIVESTOCK"
## [45] "INCRMT"         "INCWAGE"        "INCBONSOCALL"   "INCFARMBSN"
## [49] "INCNFARMBSN"    "INCRENT"        "INCFIN"         "INCPENSN"
## [53] "INCOTHER"       "INCTOTGROSSHH" "FARMEMP"        "THOUSEXP"
## [57] "TFOODEXP"       "TALCHEXP"       "TCLTHEXP"       "TFURNEXP"
## [61] "THLTHEXP"       "TTRANSEXP"      "TCOMMEXP"       "TRECEXP"
## [65] "TEDUEXP"        "TRESTHOTEXP"    "TMISCEXP"       "TANHHEXP"
```

The dataset has 10,574 individuals in 2,000 households and contains 68 variables. The survey corresponds to a population of about 4.3 million individuals, which means that the sample is relatively small and the sample weights are high. This has an impact on the disclosure risk, as we will see in Steps 6a and 6b.

To get an overview of the values of the variables, we use tabulations and cross-tabulations for categorical variables and summary statistics for continuous variables. To include the number of missing values (NA or other), we use the option useNA = "ifany" in the table() function (see Listing 12.4).

In Table 12.1 the variables in the dataset are listed along with concise descriptions of the variables, the level at which they are collected (individual (IND), household (HH)), the measurement type (continuous, semi-continuous, categorical) and value ranges. Note that the dataset contains a selection of 68 variables (cf. Appendix A) of a total of 112 variables in the survey dataset. The variables have been preselected based on their relevance for data users. This allows to reduce the total numbers of variables to consider in the anonymization process and makes the process easier. The numerical values for many of the categorical variables are codes that refer to values, e.g., in the variable URBRUR, 1 stands for rural and 2 for urban. More information on the meanings of coded values of the categorical variables is available in the *R* code for this case study.

We identified the following sensitive variables in the data: ETHNICITY, RELIGION, variables related to the labor force status of the individual and the variables containing information on income and expenditures of the household. Whether variables can be identified as sensitive may vary across countries and datasets.

The case study dataset does not have any direct identifiers that, if they were present, would need to be removed at this stage. Examples of direct identifiers would be names, telephone numbers, geographical location coordinates, etc.

Listing 12.4: Tabulation of the variable 'gender' and summary statistics for the variable 'total annual expenditures' in *R*

```
# tabulation of variable GENDER (sex, categorical)
table(file$GENDER, useNA = "ifany")
##    0    1
## 5448 5126

# summary statistics for variable TANHHEXP (total annual household expenditures,
# continuous)
summary(file$TANHHEXP)
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     498   15550   17290   28560   29720  353200
```

Table 12.1: Overview of variables in dataset

No.	Variable name	Description	Level	Measurement	Va
1	IDH	Household ID	HH	.	1-
2	IDP	Individual ID	IND	.	1-

Table 12.1 – continued from previous page

No.	Variable name	Description	Level	Measurement	Va
3	REGION	Region	HH	categorical	1-
4	DISTRICT	District	HH	categorical	10
5	URBRUR	Area of residence	HH	categorical	1,
6	WGTHH	Individual weighting coefficient	HH	weight	31
7	WGTPOP	Population weighting coefficient	IND	weight	45
8	HHSIZE	Household size	HH	semi-cont	1-
9	GENDER	Gender	IND	categorical	0,
10	REL	Relationship to household head	IND	categorical	1-
11	MARITAL	Marital status	IND	categorical	1-
12	AGEYRS	Age in completed years	IND	semi-continuous	0-
13	AGEMTH	Age of child in completed years	IND	semi-continuous	1-
14	RELIG	Religion of household head	HH	categorical	1,
15	ETHNICITY	Ethnicity of household head	HH	categorical	all
16	LANGUAGE	Language of household head	HH	categorical	all
17	MORBID	Morbidity last x weeks	IND	categorical	0,
18	MEASLES	Child immunized against measles	IND	categorical	0,
19	MEDATT	Sought medical attention	IND	categorical	0,
20	CHWEIGHTKG	Weight of child (Kg)	IND	continuous	2 -
21	CHHEIGHTCM	Height of child (cms)	IND	continuous	7 -
22	ATSCHOOL	Currently enrolled in school	IND	categorical	0,
23	EDUCY	Highest level of education attended	IND	categorical	1-
24	EDYEARS	Years of education	IND	semi-continuous	0-
25	EDYRSCURRAT	Years of education for currently enrolled	IND	semi-continuous	1-
26	SCHTYP	Type of school attending	IND	categorical	1-
27	LITERACY	Literacy	IND	categorical	1-
28	EMPTYP1	Type of employment	IND	categorical	1-
29	UNEMP1	Unemployed	IND	categorical	0,
30	INDUSTRY1	Industry classification (1-digit)	IND	categorical	1-
31	EMPCAT1	Employment categories	IND	categorical	11
32	WHOURSLASTWEEK1	Hours worked last week	IND	continuous	0-
33	OWNHOUSE	Ownership of dwelling	HH	categorical	0,
34	ROOF	Main material used for roof	IND	categorical	1-
35	TOILET	Main toilet facility	HH	categorical	1-
36	ELECTCON	Electricity	HH	categorical	0-
37	FUELCOOK	Main cooking fuel	HH	categorical	1-
38	WATER	Main source of water	HH	categorical	1-
39	OWNAGLAND	Ownership of agricultural land	HH	categorical	1-
40	LANDSIZEHA	Land size owned by household (ha) (agric and non agric)	HH	continuous	0-
41	OWNMOTORCYCLE	Ownership of motorcycle	HH	categorical	0,
42	CAR	Ownership of car	HH	categorical	0,
43	TV	Ownership of television	HH	categorical	0,
44	LIFESTOCK	Number of large-sized livestock owned	HH	semi-continuous	0-
45	INCRMT	Income – Remittances	HH	continuous	
46	INCWAGE	Income - Wages and salaries	HH	continuous	
47	INCBONSOCAL	Income - Bonuses and social allowances derived from wage jobs	HH	continuous	
48	INCFARMBSN	Income - Gross income from household farm businesses	HH	continuous	
49	INCNFARMBSN	Income - Gross income from household nonfarm businesses	HH	continuous	
50	INCRENT	Income - Rent	HH	continuous	
51	INCFIN	Income - Financial	HH	continuous	

Table 12.1 – continued from previous page

No.	Variable name	Description	Level	Measurement	Va
52	INCPENSN	Income - Pensions/social assistance	HH	continuous	
53	INCOTHER	Income - Other	HH	continuous	
54	INCTOTGROSHH	Income - Total	HH	continuous	
55	FARMEMP				
56	TFOODEXP	Total expenditure on food	HH	continuous	
57	TALCHEXP	Total expenditure on alcoholic beverages, tobacco and narcotics	HH	continuous	
58	TCLTHEXP	Total expenditure on clothing	HH	continuous	
59	THOUSEXP	Total expenditure on housing	HH	continuous	
60	TFURNEXP	Total expenditure on furnishing	HH	continuous	
61	THLTHEXP	Total expenditure on health	HH	continuous	
62	TTRANSEXP	Total expenditure on transport	HH	continuous	
63	TCOMMEXP	Total expenditure on communication	HH	continuous	
64	TRECEXP	Total expenditure on recreation	HH	continuous	
65	TEDUEXP	Total expenditure on education	HH	continuous	
66	TRESHOTEXP	Total expenditure on restaurants and hotels	HH	continuous	
67	TMISCEXP	Total expenditure on miscellaneous spending	HH	continuous	
68	TANHHEXP	Total annual nominal household expenditures	HH	continuous	

It is always important to ensure that the relationships between variables in the data are preserved during the anonymization process and to explore and take note of these relationships before beginning the anonymization. In the final step in the anonymization process, an audit should be conducted, using these initial results, to check that these relationships are maintained in the anonymized dataset.

In our demo dataset, we identify several relationships between variables that need to be preserved during the anonymization process. The variables TANHHEXP and INCTOTGROSSHH represent the total annual nominal household expenditure and the total gross annual household income, respectively, and these variables are aggregations of existing income and expenditure components in the dataset.

The variables related to education are available only for individuals in the appropriate age groups and missing for other individuals. We make a similar observation for variables relating to children, such as height, weight and age in months. In addition, the household-level variables (cf. fourth column of Table 12.1) have the same values for all members in any particular household. The value of household size corresponds to the actual number of individuals belonging to that household in the dataset. As we proceed, we have to take care that these relationships and structures are preserved in the anonymization process.

When tabulating the variables, we notice that the variables RELIG, EMPTYP1 and LIVESTOCK have missing value codes different from the *R* standard missing value code NA. Before proceeding, we need to recode these to NA so *R* interprets them correctly. The missing value codes are resp. 99999, 99 and 9999 for these three variables. These are genuine missing value codes and not caused by the variables being not applicable to the individual. Listing 12.5 shows how to make these changes.

Note: At the end of the anonymization process, and if desired for users, it is relatively easy to change these values back to their original missing value code.

Listing 12.5: Recoding missing value codes

```
# Set different NA codes to R missing value NA
file[,'RELIG'][file[,'RELIG'] == 99999]        <- NA
file[,'EMPTYP1'][file[,'EMPTYP1'] == 99]       <- NA
file[,'LIVESTOCK'][file[,'LIVESTOCK'] == 9999] <- NA
```

We also take note that the variables LANGUAGE and ETHNICITY have only missing values. Variables that contain

only missing values should be removed from the dataset at this stage and excluded from the anonymization process. Removing these variables does not mean loss of data or reduction of the data utility, since these variables did not contain any information. It is, however, necessary to remove them, because keeping them can lead to errors in some of the anonymization methods in *R*. It is always possible to add these variables back into the dataset to be released at the end of the anonymization process. It is useful to reduce the dataset to those variables and records relevant for the anonymization process. This guarantees the best results in *R* and fewer errors. In Listing 12.6 we drop the variables that contain all missing values.

Listing 12.6: Dropping variables with only missing values

```
# Drop variables containing only missings
file <- file[,!names(file) %in% c('LANGUAGE', 'ETHNICITY')]
```

We assume that the data are collected in a survey that uses simple sampling of households. The data contains two weight coefficients: WGTHH and WGTPOP. The relationship between the weights is WGTPOP = WGTHH * HHSIZE. WGTPOP is the sampling weight for the households and WGTHH is the sampling weight for the individuals to be used for disclosure risk calculations. WGTHH is used for computing individual-level indicators (such as education) and WGTPOP is used for population level indicators (such as income indicators). There are no strata variables available in the data. We will use WGTPOP for the anonymization of the household variables and WGTHH for the anonymization of the individual-level variables.

12.1.3 Step 3: Type of release

In this case study, we assume that data will be released as a SUF, which will be only available under license to accredited researchers with approved research proposals (see the Section Conditions for SUFs for more information of the release of a SUF). Therefore, the accepted risk level is higher and a broader set of variables can be released than would be the case when releasing a PUF. Since we do not have an overview of the requirements of all users, we restrict the utility measures to a selected number of data uses (see Step 5).

12.1.4 Step 4: Intruder scenarios and choice of key variables

Next, we analyze possible intruder scenarios and select quasi-identifiers or key variables based on these scenarios. Since the dataset used in this case study is a demo dataset that does not stem from an existing country (and hence we do not have information on external data sources available to possible intruders) and the original data has already been anonymized, it is not possible to define exact disclosure scenarios. Instead, we draft intruder scenarios for this demo dataset based on some hypothetical assumptions about availability of external data sources. We consider two types of disclosure scenarios: 1) matching to other publicly available datasets and 2) spontaneous recognition. The license under which the dataset will be distributed (SUF) prohibits matching to external resources. Still this can happen. However, the more important scenario is the one of spontaneous recognition. We describe both scenarios in the following two paragraphs.

For the sake of illustration, we assume that population registers are available with the demographic variables gender, age, place of residence (region, urban/rural), religion and other variables such as marital status and variables relating to education and professional status that are also present in our dataset. In addition, we assume that there is a publically available cadastral register on land ownership. Based on this analysis of available data sources, we select the variables REGION, URBRUR, HHSIZE, OWNAGLAND, RELIG, GENDER, REL (relationship to household head), MARITAL (marital status), AGEYRS, INDUSTRY1 and two variables relating to school attendance as categorical quasi-identifiers, the expenditure and income variables as well as LANDSIZEHA as continuous quasi-identifiers. According to our assessment, these variables might enable an intruder to re-identify an individual or household in the dataset by matching with other available datasets.

Table 12.2 gives an overview of the selected quasi-identifiers and their levels of measurement.

The decision to release the dataset as a SUF means the level of anonymization will be relatively low and consequently, the variables are more detailed and a scenario of spontaneous recognition is our main concern. Therefore, we should check for rare combinations or unusual patterns in the variables. Variables that may lead to spontaneous recognition in our sample are amongst others HHSIZE (household size), LANDSIZEHA as well as income and expenditure variables. Large households and large land ownership are easily identifiable. The same holds for extreme outliers in wealth and expenditure variables, especially when combined with other identifying variables such as region. There might be only one or a few households in a certain region with a high income, such as the local doctor. Variables that are easily observable and known by neighbors such as ROOF, TOILET, WATER, ELECTCON, FUELCOOK, OWNMOTORCYCLE, CAR, TV and LIVESTOCK may also need protection depending on what stands out in the community, since a researcher might be able to identify persons (s)he knows. This is called the nosy-neighbor scenario.

Table 12.2: List of selected quasi-identifiers

Name	Measurement
REGION (region)	Household, categorical
URBRUR (area of residence)	Household, categorical
HHSIZE (household size)	Household, categorical
OWNAGLAND (agricultural land ownership)	Household, categorical
RELIG (religion of household head)	Household, categorical
LANDSIZEHA (size of agr. and non-agr. land)	Household, continuous
TANHHEXP (total expenditures)	Household, continuous
TEXP (expenditures in category)	Household, continuous
INCTOTGROSSHH (total income)	Household, continuous
INC (income in category)	Household, continuous
GENDER (sex)	Individual, categorical
REL (relationship to household head)	Individual, categorical
MARITAL (marital status)	Individual, categorical
AGEYRS (age in completed years)	Individual, semi-continuous
EDYRSCURATT (years of education for currently enrolled)	Individual, semi-continuous
EDUCY (highest level of education completed)	Individual, categorical
ATSCHOOL (currently enrolled in school)	Individual, categorical
INDUSTRY1 (industry classification)	Individual, categorical

12.1.5 Step 5: Data key uses and selection of utility measures

In this case study, our aim is to create a SUF that provides sufficient information for accredited researchers. We know that the primary use of these data will be to evaluate indicators relating to income and inequality. Examples are the GINI coefficient and indicators on what share of income is spent on what type of expenditures. Furthermore, we focus on some education indicators. Table 12.3 gives an overview of the utility measures we selected. Besides these utility measures, which are specific to the data uses, we also do standard checks, such as comparing tabulations, cross-tabulations and summary statistics before and after anonymization.

Table 12.3: Overview of selected utility measures

Gini point estimates and confidence intervals for total expenditures	.
Lorenz curves for total expenditures	
Mean monthly per capita total expenditures by area of residence	
Average share of components for expenditures	
Mean monthly per capita total income by area of residence	
Average share of components for income	
Net enrollment in primary education by gender	

There are no published figures and statistics available that are calculated from this dataset because it is a demo. In

general, the published figures should be re-computed based on the anonymized dataset and compared to the published figures in Step 11. Large differences would reduce the credibility of the anonymized dataset.

12.1.6 Hierarchical (household) structure

Our demo survey collects data on individuals in households. The household structure is important for data users and should be considered in the risk assessment. Since some variables are measured on the household level and thus have identical values for each household member, the values of the household variables should be treated in the same way for each household member (see the Section Anonymization of the quasi-identifier household size). Therefore, we first anonymize only the household variables. After this, we merge them with the individual-level variables and then anonymize the individual-level and household-level variables jointly.

Since the data has a hierarchical structure, Steps 6 through 10 are repeated twice: Steps 6a through 10a are for the household-level variables and Steps 6b through 10b for the combined dataset. In this way, we ensure that household-level variable values remain consistent across household members for each household and the household structure cannot be used to re-identify individuals. This is further explained in the Sections Levels of risk and Randomizing order and numbering of individuals or households .

Before continuing to Step 6a, we select the categorical key variables, continuous key variables and any variables selected for use in PRAM routines, as well as household-level sampling weights. We extract these selected household variables and the households from the dataset and save them as *fileHH*. The choice of PRAM variables is further explained in Step 8a. Listing 12.7 illustrates how these steps are done in *R* (see also the Section Household structure).

Note: In our dataset, some of the categorical variables when imported from the STATA file were not imported as factors. sdcMicro requires that these be converted to factors before proceeding.

Conversion of these variables to factors is also shown in Listing 12.7.

Listing 12.7: Selecting the variables for the household-level anonymization

```
### Select variables (household level)
# Key variables (household level)
selectedKeyVarsHH = c('URBRUR', 'REGION', 'HHSIZE', 'OWNHOUSE',
                      'OWNAGLAND', 'RELIG')

# Changing variables to class factor
file$URBRUR    <- as.factor(file$URBRUR)
file$REGION    <- as.factor(file$REGION)
file$OWNHOUSE  <- as.factor(file$OWNHOUSE)
file$OWNAGLAND <- as.factor(file$OWNAGLAND)
file$RELIG     <- as.factor(file$RELIG)

# Numerical variables
numVarsHH = c('LANDSIZEHA', 'TANHHEXP',   'TFOODEXP',       'TALCHEXP',
              'TCLTHEXP',   'THOUSEXP',   'TFURNEXP',       'THLTHEXP',
              'TTRANSEXP',  'TCOMMEXP',   'TRECEXP',        'TEDUEXP',
              'TRESHOTEXP', 'TMISCEXP',   'INCTOTGROSSHH', 'INCRMT',
              'INCWAGE',    'INCFARMBSN', 'INCNFARMBSN',    'INCRENT',
              'INCFIN',     'INCPENSN',   'INCOTHER')
# PRAM variables
pramVarsHH = c('ROOF', 'TOILET', 'WATER', 'ELECTCON',
               'FUELCOOK', 'OWNMOTORCYCLE', 'CAR', 'TV', 'LIVESTOCK')

# sample weight (WGTPOP) (household)
```

(continues on next page)

(continued from previous page)

```
weightVarHH = c('WGTPOP')

# All household level variables
HHVars <- c('HID', selectedKeyVarsHH, pramVarsHH, numVarsHH, weightVarHH)
```

We then extract these variables from *file*, the dataframe in *R* that contains all variables. Every household has the same number of entries as it has members (e.g., a household of three will be repeated three times in *fileHH*). Before analyzing the household-level variables, we select only one entry per household, as illustrated in Listing 12.8. This is further explained in the Section Household structure.

Listing 12.8: Taking a subset with only households

```
# Create subset of file with households and HH variables
fileHH <- file[,HHVars]

# Remove duplicated rows based on IDH, select uniques,
# one row per household in fileHH
fileHH <- fileHH[which(!duplicated(fileHH$IDH)),]

dim(fileHH)
## [1] 2000   39
```

The file *fileHH* contains 2,000 households and 39 variables. We are now ready to create our *sdcMicro* object with the corresponding variables we selected in Listing 12.7. For our case study, we will create an *sdcMicro* object called *sdcHH* based on the data in *fileHH*, which we will use for steps 6a – 10a (see Listing 12.9).

Note: When the sdcMicro object is created, the sdcMicro package automatically calculates and stores the risk measures for the data.

This leads us to Step 6a.

Listing 12.9: Creating a *sdcMicro* object for the household variables

```
# Create initial SDC object for household level variables
sdcHH <- createSdcObj(dat = fileHH, keyVars = selectedKeyVarsHH, pramVars =␣
↪pramVarsHH,
                      weightVar = weightVarHH, numVars = numVarsHH)

numHH <- length(fileHH[,1]) # number of households
```

12.1.7 Step 6a: Assessing disclosure risk (household level)

As a first measure, we evaluate the number of households violating k-anonymity at the levels 2, 3 and 5.

Table 12.4 shows the number of violating households as well as the percentage of the total number of households. Listing 12.10 illustrates how to find these values with *sdcMicro*. The print() function in *sdcMicro* shows only the values for thresholds 2 and 3. Values for other thresholds can be calculated manually by summing up the frequencies smaller than the k-anonymity threshold, as shown in Listing 12.10.

Table 12.4: Number and proportion of households violating k-anonymity

k-anonymity level	Number of HH violating	Percentage of total number of HH
2	103	5.15 %
3	229	11.45 %
5	489	24.45 %

Listing 12.10: Showing number of households violating k-anonymity for levels 2, 3 and 5

```
# Number of observations violating k-anonymity (thresholds 2 and 3)
print(sdcHH)
## Infos on 2/3-Anonymity:
##
## Number of observations violating
##   - 2-anonymity: 103
##   - 3-anonymity: 229
##
## Percentage of observations violating
##   - 2-anonymity: 5.150 %
##   - 3-anonymity: 11.450 %
-----------------------------------------------------------------------

# Calculate sample frequencies and count number of obs. violating k(5) - anonymity
kAnon5 <- sum(sdcHH@risk$individual[,2] < 5)

kAnon5
## [1] 489

# As percentage of total
kAnon5 / numHH
## [1] 0.2445
```

It is often useful to view the values for the household(s) that violate k-anonymity. This might help clarify which variables cause the uniqueness of these households; this can then be used later when choosing appropriate SDC methods. Listing 12.11 shows how to assess the values of the households violating 3- and 5-anonymity. It seems that among the categorical key variables, the variable HHSIZE is responsible for many of the unique combinations and the origin of much of the risk. Having determined this, we can flag HHSIZE as a possible first variable to treat to obtain the required risk level. In practice, with a variable like HHSIZE, this will likely involve removing large households from the dataset to be released. As explained in the Section Anonymization of the quasi-identifier household size , recoding and local suppression are no valid options for the variable HHSIZE. The frequencies of household size in Table 12.7 show that there are few households with more than 13 household members. This makes these households easily identifiable based on the number of household members and at high risk of re-identification, also in the context of the nosy neighbor scenario.

Listing 12.11: Showing households that violate k-anonymity

```
# Show values of key variable of records that violate k-anonymity
fileHH[sdcHH@risk$individual[,2] < 3, selectedKeyVarsHH] # for 3-anonymity
fileHH[sdcHH@risk$individual[,2] < 5, selectedKeyVarsHH] # for 5-anonymity
```

We also assess the disclosure risk of the categorical variables with the individual and global risk measures as described in the Sections Individual risk and Global risk. In *fileHH* every entry represents a household. Therefore, we use the individual non-hierarchical risk here, where the individual refers in this case to a household. *fileHH* contains only households and has no hierarchical structure. In Step 6b, we evaluate the hierarchical risk in *file*, the dataset containing both households and individuals. The individual and global risk measures automatically take into consideration the

household weights, which we defined in Listing 12.7. In our file, the global risk measure calculated using the chosen key variables is 0.05%. This percentage is extremely low and corresponds to 1.03 expected re-identifications. The results are also shown in Listing 12.12. This low figure can be explained by the relatively small sample size of 0.25% of the total population. Furthermore, one should keep in mind that this risk measure is based only on the categorical quasi-identifiers at the household level. Listing 12.12 illustrates how to print the global risk measure.

Listing 12.12: Printing global risk measures

```
print(sdcHH, "risk")

## Risk measures:
##
## Number of observations with higher risk than the main part of the data: 0
## Expected number of re-identifications: 1.03 (0.05 %)
```

The global risk measure does not provide information about the spread of the individual risk measures. There might be a few households with relatively high risk, while the global (average) risk is low. It is therefore useful as a next step to inspect the observations with relatively high risk. The highest risk is 5.5% and only 14 households have risk larger than 1%. Listing 12.13 shows how to display those households with risk over a certain threshold. Here the threshold is 0.01 (1%).

Listing 12.13: Observations with individual risk higher than 1%

```
# Observations with risk above certain threshold (0.01)
fileHH[sdcHH@risk$individual[, "risk"] > 0.01,]
```

Since the selected key variables at the household level are both categorical and numerical, the individual and global risk measures based on frequency counts do not completely reflect the disclosure risk of the entire dataset. Both categorical and continuous key variables are important for the data users, thus options like recoding the continuous variables (e.g., by creating quantiles of income and expenditure variables) to make all of them categorical will likely not satisfy the data user's needs. We therefore avoid recoding continuous variables and assess the disclosure risk of the categorical and continuous variables separately. This approach can be partly justified by the fact that any potential matching to external data sources for the continuous and categorical variables are available from different external data sources and as such will not be used simultaneously for matching.

Continuous variables

To measure the risk of the continuous variables, we use an interval measure, which measures the number of anonymized values that are too close to their original values. See the Section Interval measure for more information on interval-based risk measures for continuous variables. This measure is an ex-post measure, meaning that the risk can be evaluated only after anonymization and measures whether the perturbation is sufficiently large. Since it is an ex-post measure, we can evaluate it only in Step 9a after the variables have been treated. Evaluating this measure based on the original data would lead to a risk of 100%; all values would be too close to the original values since they would coincide with the original values, no matter how small the chosen intervals would be.

We also look at the distribution of LANDSIZEHA. In the variable LANDSIZEHA high values are rare and can lead to re-identification. An example is a large landowner in a specific region. To evaluate the distribution of the variable LANDSIZEHA, we look at the percentiles. Every percentile represents approximately 20 households. In addition, we look at the values of the largest 50 plots. Listing 12.14 shows how to use *R* to display the quantiles and the largest landplots. Table 12.5 shows the 90^{th} – 100^{th} percentiles and Table 12.6 displays the largest 50 values for LANDSIZEHA. Based on these values, we conclude that values of LANDSIZEHA over 40 make the household very identifiable. These large households and households with large land plots need extra protection, as discussed in Step 8a.

Listing 12.14: Percentiles of LANDSIZE and listing the sizes of the largest 50 plots

```
# 1st - 100th percentiles of land size
quantile(fileHH$LANDSIZEHA, probs = (1:100)/100, na.rm= TRUE)

# Values of landsize for largest 50 plots
tail(sort(fileHH$LANDSIZEHA), n = 50)
```

Table 12.5: Percentiles 90-100 of the variable LANDSIZE

Percentile	90	91	92	93	94	95
Value	6.00	8.00	8.09	10.12	10.12	10.12
Percentile	96	97	98	99	100	
Value	12.14	20.23	33.83	121.41	1,214.08	

Table 12.6: 50 largest values of the variable LANDSIZE

12.14	15.00	15.37	15.78	16.19	20.00	20.23	20.23	20.23	20.23
20.23	20.23	20.23	20.23	20.23	20.23	20.23	20.23	20.23	20.23
20.23	20.23	20.50	30.35	32.38	40.47	40.47	40.47	40.47	40.47
40.47	40.47	80.93	80.93	80.93	80.93	121.41	121.41	161.88	161.88
161.88	182.11	246.86	263.05	283.29	404.69	404.69	607.04	809.39	1214.08

12.1.8 Step 7a: Assessing utility measures (household level)

The utility of the data does not only depend on the household level variables, but on the combination of household-level and individual-level variables. Therefore, it is not useful to evaluate all the utility measures selected in Step 5 at this stage, i.e., before anonymizing the individual level variables. We restrict the initial measurement of utility to those measures that are solely based on the household variables. In our dataset, these are the measures related to income and expenditure and their distributions. The results are presented in Step 10a, together with the results after anonymization, which allow direct comparison. If after the next anonymization step it appears that the data utility has been significantly decreased by the suppression of some household level variables, we can return to this step.

12.1.9 Step 8a: Choice and application of SDC methods (household variables)

This step is divided into the anonymization of the variable HHSIZE, as this is a special case, the anonymization of the other selected categorical quasi-identifiers and the anonymization of the selected continuous quasi-identifiers.

Variable HHSIZE

The variable HHSIZE poses a problem for the anonymization of the file, since suppressing it will not anonymize this variable: a simple headcount based on the household ID would allow the reconstruction of this variable. Table 12.7 shows the absolute frequencies of HHSIZE. The number of households for each size larger than 13 is 6 or fewer and can be considered outliers with a higher risk of re-identification, as discussed in Step 6a. One way to deal with this is to remove all households of size 14 or larger from the dataset[1]. Removing 29 households of size 14 or larger reduces the number of 2-anonymity violations by 18, of 3-anonymity violations by 26 and of 5-anonymity violations by 29. This means that all removed households violated 5-anonymity due to the value of the variable HHSIZE and many of them 2- or 3-anonymity. In addition, the average individual risk amongst the 29 households is 0.15%, which is almost three times higher than the average individual risk of all households. The impact on the global risk measure of removing

[1] Other methods and guidance on treating datasets where household size is a quasi-identifier are discussed in the Section Anonymization of the quasi-identifier household size.

these 29 households is, however, limited, due to the relatively small number of removed households in comparison to the total number of 2,000 households. Removing the households is primarily to protect these specific households, not to reduce the global risk.

Changes, such as removing records, cannot be done in the *sdcMicro* object. Listing 12.15 illustrates the way to remove households and recreate the *sdcMicro* object.

Table 12.7: Frequencies of variable HHSIZE (household size)

HHSIZE	1	2	3	4	5	6	7	8	9	10	11	12
Frequency	152	194	238	295	276	252	214	134	84	66	34	21
HHSIZE	13	14	15	16	17	18	19	20	21	22	33	
Frequency	11	6	6	5	4	2	1	2	1	1	1	

Listing 12.15: Removing households with large (rare) household sizes

```
# Tabulation of variable HHSIZE
table(sdcHH@manipKeyVars$HHSIZE)

# Remove large households (14 or more household members) from file and fileHH
file <- file[!file[,'HHSIZE'] >= 14,]

fileHHnew <- fileHH[!fileHH[,'HHSIZE'] >= 14,]

# Create new sdcMicro object based on the file without the removed households
sdcHH <- createSdcObj(dat=fileHHnew, keyVars=selectedKeyVarsHH, pramVars=pramVarsHH,
                      weightVar=weightVarHH, numVars = numVarsHH)
```

Categorical variables

We are now ready to move on to the choice of SDC methods for the categorical variables on the household level in our dataset. As noted in our discussion of the methods, applying perturbative methods and local suppression may lead to large loss of utility. The common approach is to apply recoding to the largest possible extent as a first approach, to reach a prescribed level of risk and reduce the number of suppressions needed. Only after that should methods such as local suppression be applied. If this approach does not already achieve the desired result, we can consider perturbative methods.

Since the file is to be released as a SUF, we can keep a higher level of detail in the data. The selected categorical key variables at the household level are not suitable for recoding at this point. Due to the relatively low risk of re-identification based on the five selected categorical household level variables, it is possible in this case to use an option like local suppression to achieve our desired level of risk. Applying local suppression when initial risk is relatively low will likely only lead to suppression of few observations and thus limit the loss of utility. If, however, the data had been measured to have a relatively high risk, then applying local suppression without previous recoding would likely result in a large number of suppressions and greater information loss. Efforts such a recoding should be taken first before suppressing in cases where risk is initially measured as high. Recoding will reduce risk with little information loss and thus the number of suppressions, if local suppression is applied as an additional step. We apply local suppression to reach 2-anonymity. The choice of the low level of two is based on the overall low re-identification risk due to the high sample weights and the release as SUF. High sample weights mean, ceteris paribus, a low level of re-identification risk. Achieving 2-anonymity is the same as removing sample uniques. This leads to 42 suppressions in the variable HHSIZE and 4 suppressions in the variable REGION. As explained earlier, suppression of the value of the variable HHSIZE does not lead to actual suppression of this information. Therefore, we redo the local suppression, but this time we tell *sdcMicro* to, if possible, not suppress HHSIZE but one of the other variables.

In *sdcMicro* it is possible to tell the algorithm which variables are important and less important for making small changes (see also the Section Local suppression). To prevent HHSIZE being suppressed, we set the importance of HHSIZE in the importance vectors to the highest (i.e., 1). Listing 12.16 shows how to apply local suppression and put importance on the variable HHSIZE. The variable REGION is the type of variable that should not have any

suppressions either. We also set the importance of REGION to 2 and the importance of RURURB to 3. This leads to an order of the variables to be considered for suppression by the algorithm. Instead of 42 suppressions in the variable HHSIZE, this leads one suppressed value in the variable HHSIZE, and to 6, 1, 48 and 16 suppressions respectively for the variables URBRUR, REGION, OWNAGLAND and RELIG (which we set as less important). The importance is clearly reflected in the number of suppression. The total number of suppressions is higher than without importance vector (71 vs. 46), but 2-anonymity is achieved in the dataset with fewer suppressions in the variables HHSIZE and REGION. We remove the one household with the suppressed value of HHSIZE (13) to protect this household.

Note: In Listing 12.16 we use the undolast() function in sdcMicro to go one step back after we had first applied local suppression with no importance vector.

The undolast() function restores the *sdcMicro* object back to the previous state (i.e., before we applied local suppression), which allows us to rerun the same command, but this time with an importance vector set. The undolast() function can only be used to go one step back.

Listing 12.16: Local suppression with and without importance vector

```
# Local suppression
sdcHH <- localSuppression(sdcHH, k=2, importance = NULL) # no importance vector

print(sdcHH, "ls")
## Local Suppression:
##     KeyVar | Suppressions (#) | Suppressions (%)
##     URBRUR |                 0 |            0.000
##     REGION |                 4 |            0.203
##     HHSIZE |                37 |            1.877
##  OWNAGLAND |                 0 |            0.000
##      RELIG |                 0 |            0.000

sdcHH <- undolast(sdcHH)

sdcHH <- localSuppression(sdcHH, k=2, importance = c(3, 2, 1, 5, 5))
# importance on HHSIZE (1), REGION (2) and URBRUR (3)

print(sdcHH, "ls")
## Local Suppression:
##     KeyVar | Suppressions (#) | Suppressions (%)
##     URBRUR |                 6 |            0.304
##     REGION |                 1 |            0.051
##     HHSIZE |                 1 |            0.051
##  OWNAGLAND |                43 |            2.182
##      RELIG |                16 |            0.812
```

The variables ROOF, TOILET, WATER, ELECTCON, FUELCOOK, OWNMOTORCYCLE, CAR, TV and LIVESTOCK are not sensitive variables and were not selected as quasi-identifiers because we assumed that there are no external data sources containing this information that could be used for matching. Values can be easily observed or be known to neighbors, however, and therefore are important, together with other variables, for the spontaneous recognition scenario and nosy neighbor scenario. To anonymize these variables, we want to introduce a low level of uncertainty in them. Therefore, we decide to use invariant PRAM for the variables ROOF, TOILET, WATER, ELECTCON, FUELCOOK, OWNMOTORCYCLE, CAR, TV and LIVESTOCK, where we treat LIVESTOCK as a semi-continuous variable due to the low number of different values. The Section PRAM (Post RAndomization Method) provides more information on the PRAM method and its implementation in *sdcMicro*. Listing 12.17 illustrates how to apply PRAM. We choose the parameter *pd*, the lower bound for the probability that a value is not changed, to be relatively high at 0.8. We can choose a high value, because the variables themselves are not sensitive and we only want to introduce a low level of changes to minimize the utility loss. Because the distribution of many of

the variables chosen for PRAM depends on the REGION, we decide to use the variable REGION as a strata variable. In this way the transition matrix is computed for each region separately. Because PRAM is a probabilistic method, we set a seed for the random number generator before applying PRAM to ensure reproducibility of the results.

Note: In practice, it is not advisable to set a seed of 12345, but rather a longer more complex and less easy to guess sequence.

The seed should not be released, since it allows for reconstructing the original values if combined with the transition matrix. The transition matrix can be released: this allows for consistent statistical inference by correcting the statistical methods used if the researcher has knowledge about the PRAM method (at this point *sdcMicro* does not allow the retrieval of the transition matrix).

Listing 12.17: Applying PRAM

```
# Pram
set.seed(12345)
sdcHH <- pram(sdcHH, strata_variables = "REGION", pd = 0.8)

## Number of changed observations:
## - - - - - - - - - - -
## ROOF != ROOF_pram : 98 (4.97%)
## TOILET != TOILET_pram : 151 (7.66%)
## WATER != WATER_pram : 167 (8.47%)
## ELECTCON != ELECTCON_pram : 90 (4.57%)
## FUELCOOK != FUELCOOK_pram : 113 (5.73%)
## OWNMOTORCYCLE != OWNMOTORCYCLE_pram : 41 (2.08%)
## CAR != CAR_pram : 172 (8.73%)
## TV != TV_pram : 137 (6.95%)
## LIVESTOCK != LIVESTOCK_pram : 149 (7.56%)
```

PRAM has changed values within the variables according to the invariant transition matrices. Since we used the invariant PRAM method (see the Section PRAM (Post RAndomization Method)), the absolute univariate frequencies remain unchanged. This is not the case for the multivariate frequencies. In Step 10a we compare the changes in the multivariate frequencies for the PRAMmed variables.

Continuous variables

We have selected income and expenditures variables and the variable LANDSIZEHA as numerical quasi-identifiers, as discussed in Step 4. In Step 5 we identified variables having high interest for the users of our data: many users use the data for measuring inequality and expenditure patterns.

Based on the risk evaluation in Step 6a, we decide to anonymize the variable LANDSIZEHA by top coding at the value 40 (cf. Table 12.5 and Table 12.6) and round values smaller than 1 to one digit, and values larger than 1 to zero digits. Rounding the values prevents exact matching with the available cadastral register. Furthermore, we group the values between 5 and 40 in the groups 5 – 19 and 20 – 39. After these steps, no household has a unique plot size and the number of households in the sample with the same plot size was increased to at least 7. This is shown by the tabulation of the variable LANDSIZEHA after manipulation in the last line of Listing 12.18. In addition, all outliers have been removed by top coding the values. This has reduced the risk of spontaneous recognition as discussed in Step 6. How to recode values in *R* is introduced in the Section Recoding and, for this particular case, shown in Listing 12.18.

Listing 12.18: Anonymizing the variable LANDSIZEHA

```
# Rounding values of LANDSIZEHA to 1 digit for plots smaller than 1 and
# to 0 digits for plots larger than 1
sdcHH@manipNumVars$LANDSIZEHA[sdcHH@manipNumVars$LANDSIZEHA <= 1 &
```

(continues on next page)

(continued from previous page)

```
                                !is.na(sdcHH@manipNumVars$LANDSIZEHA)] <-
              round(sdcHH@manipNumVars$LANDSIZEHA[sdcHH@manipNumVars$LANDSIZEHA <= 1 &
                                                  !is.na(sdcHH@manipNumVars
↪$LANDSIZEHA)],
                    digits = 1)

sdcHH@manipNumVars$LANDSIZEHA[sdcHH@manipNumVars$LANDSIZEHA > 1 &
                                !is.na(sdcHH@manipNumVars$LANDSIZEHA)] <-
              round(sdcHH@manipNumVars$LANDSIZEHA[sdcHH@manipNumVars$LANDSIZEHA > 1 &
                                                  !is.na(sdcHH@manipNumVars
↪$LANDSIZEHA)],
                    digits = 0)

# Grouping values of LANDSIZEHA into intervals 5-19, 20-39
sdcHH@manipNumVars$LANDSIZEHA[sdcHH@manipNumVars$LANDSIZEHA >= 5 &
sdcHH@manipNumVars$LANDSIZEHA < 20 & !is.na(sdcHH@manipNumVars$LANDSIZEHA)] <- 13

sdcHH@manipNumVars$LANDSIZEHA[sdcHH@manipNumVars$LANDSIZEHA >= 20 &
sdcHH@manipNumVars$LANDSIZEHA < 40 &!is.na(sdcHH@manipNumVars$LANDSIZEHA)] <- 30

# Topcoding values of LANDSIZEHA larger than 40 (also recomputes risk after manual␣
↪changes)
sdcHH <- topBotCoding(sdcHH, value = 40, replacement = 40, kind = 'top', column =
↪'LANDSIZEHA')

# Results for LANDSIZEHA
table(sdcHH@manipNumVars$LANDSIZEHA)
##   0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9   1   2   3   4  13  30  40
## 188 109  55  30  24  65  22   7  31  16 154 258  53  60 113  18  25
```

For the expenditure and income variables we compared, **based on the actual case study data**, several methods. As mentioned earlier, the main use of the data is to compute inequality measures, such as the Gini coefficient. Recoding these variables into percentiles creates difficulties computing these measures or changes these measures to a large extent and is hence not a suitable method. Often, income and expenditure variables that are released in a SUF are anonymized by top-coding. This protects the outliers, which are the values that are the most at risk. Top-coding, however, destroys the inequality information in the data, by removing high (and low) incomes. Therefore, we decide to use noise addition. To take into account the higher risk of outliers, we add a higher level of noise to those.

Adding noise can lead to a transformation of the shape of the distribution. Depending on the magnitude of the noise (see the Section Noise addition for the definition of the magnitude of noise), the values of income can also become negative. One way to solve this would be to cut off the values below zero and set them to zero. This would, however, destroy the properties conserved by noise addition (amongst others the value of the expected mean, see also the Section Noise addition) and we chose to keep the negative values.

As mentioned before, the aggregates of income and expenditures are the sums of the components. Adding noise to each of the components might lead to violation of this condition. Therefore, one solution is to add noise to the aggregates and remove the components. We prefer to keep the components in the data and apply noise addition to each component separately. This allows to apply a lower level of noise than when applying noise only to the aggregates. A noise level of 0.01 seems to be sufficient with extra noise of 0.05 added to the outliers. The outliers are defined by a robust Mahalanobis distance (see the Section Noise addition). After adding noise to the components, we recomputed the aggregates as the sum of the perturbed components.

Note: This result is only based on the actual case study dataset and is not necessarily true for other datasets.

The noise addition is shown in Listing 12.19. Before applying probabilistic methods such as noise addition, we set a

seed for the random number generator. This allows us to reproduce the results.

Listing 12.19: Anonymizing continuous variables

```
# Add noise to income and expenditure variables by category

# Anonymize components
compExp <- c("TFOODEXP", "TALCHEXP", "TCLTHEXP", "THOUSEXP",
             "TFURNEXP", "THLTHEXP", "TTRANSEXP", "TCOMMEXP", "TRECEXP", "TEDUEXP",
             "TRESHOTEXP", "TMISCEXP")
set.seed(123)

# Add noise to expenditure variables
sdcHH <- addNoise(noise = 0.01, obj = sdcHH, variables = compExp, method = "additive")

# Add noise to outliers
sdcHH <- addNoise(noise = 0.05, obj = sdcHH, variables = compExp, method = "outdect")

# Sum over expenditure categories to obtain consistent totals
sdcHH@manipNumVars[,'TANHHEXP'] <- rowSums(sdcHH@manipNumVars[,compExp])
compInc <- c('INCRMT', 'INCWAGE', 'INCFARMBSN', 'INCNFARMBSN',
             'INCRENT', 'INCFIN', 'INCPENSN', 'INCOTHER')

# Add noise to income variables
sdcHH <- addNoise(noise = 0.01, obj = sdcHH, variables = compInc, method = "additive")

# Add noise to outliers
sdcHH <- addNoise(noise = 0.05, obj = sdcHH, variables = compInc, method = "outdect")

# Sum over income categories to obtain consistent totals
sdcHH@manipNumVars[,'INCTOTGROSSHH'] <- rowSums(sdcHH@manipNumVars[,compInc])

# recalculate risks after manually changing values in sdcMicro object
calcRisks(sdcHH)
```

12.1.10 Step 9a: Re-measure risk

For the categorical variables, we conclude that we have achieved 2-anonymity in the data with local suppression. Only 104 households, or about 5% of the total number, violate 3-anonymity. Table 12.8 gives an overview of these risk measures. The global risk is reduced to 0.02% (expected number of re-identifications 0.36), which is extremely low. Therefore, we conclude that based on the categorical variables, the data has been sufficiently anonymized. No household has a risk of re-identification higher than 0.01 (1%). By removing households with rare values (or outliers) of the variable HHSIZE, we have reduced the risk of spontaneous recognition of these households. This reasoning can also be applied to the result of the risk of recoding the variable LANDSIZEHA and PRAMming the variables identified to be important in the nosy neighbor scenario. An intruder cannot know with certainty whether a household that he recognizes in the data is the correct household, due to the noise.

Table 12.8: Number and proportion of households violating k-anonymity after anonymization

k-anonymity	Number HH violating	Percentage
2	0	0 %
3	104	5.28 %
5	374	18.70 %

These measures refer only to the categorical variables. To evaluate the risk of the continuous variables we could use

an interval measure or closest neighbor algorithm. These risk measures are discussed in the Section Risk measures for continuous variables. We chose to use an interval measure, since exact value matching is not our largest concern based on the assumed scenarios and external data sources. Instead, datasets with similar values but not the exact same values could be used for matching. Here the main concern is that the values are sufficiently far from the original values, which is measured with an interval measure.

Listing 12.20 shows how to evaluate the interval measure for each of the expenditure variables, which are contained in the vector *compExp*[2]. The different values of the parameter *k* in the function dRisk() define the size of the interval around the original value, as explained in the Section Interval measure. The larger *k*, the larger the intervals, the higher the probability that a perturbed value is in the interval around the original value and the higher the risk measure. The result is satisfactory with relatively small intervals (k = 0.01), but not when increasing the size of the intervals. In our case, k = 0.01 is sufficiently large, since we are looking at the components, not the aggregates. We have to pay special attention to the outliers. Here the value 0.01 for k is too small to assume that they are protected when outside this small interval. It would be necessary to check outliers and their perturbed values and there might be a need for a higher level of perturbation for outliers. We conclude that, from a risk perspective and based on the interval measure, the chosen levels of noise are acceptable. In the next step, we will look at the impact on the data utility of the noise addition.

Listing 12.20: Measuring risk of re-identification of continuous variables

```
dRisk(sdcHH@origData[,compExp], xm = sdcHH@manipNumVars[,compExp], k = 0.01)
[1] 0.0608828

dRisk(sdcHH@origData[,compExp], xm = sdcHH@manipNumVars[,compExp], k = 0.02)
[1] 0.9025875

dRisk(sdcHH@origData[,compExp], xm = sdcHH@manipNumVars[,compExp], k = 0.05)
[1] 1
```

12.1.11 Step 10a: Re-measure utility

None of the variables has been recoded and the original level of detail in the data is kept, except for the variable LANDSIZEHA. As described in Step 8a, local suppression has only removed a few values in the other variables, which has not greatly reduced the validity of the data.

The univariate frequency distributions of the variables ROOF, TOILET, WATER, ELECTCON, FUELCOOK, OWNMOTORCYCLE, CAR, TV and LIVESTOCK did not, by definition of the invariant PRAM method (see the Section PRAM (Post RAndomization Method)), change to a large extent. The tabulations are presented in Table 12.9 (the values 1 – 9 and NA in the first row are the values of the variables and .m after the variable name refers to the values after anonymization).

Note: Although the frequencies are almost the same, this does not mean that the values of particular households did not change.

Values have been swapped between households. This becomes apparent when looking at the multivariate frequencies of the WATER with the variable URBRUR in Table 12.10. The multivariate frequencies of the PRAMmed with the variable URBRUR could be of interest for users, but these are not preserved. Since we applied PRAM within the regions, the multivariate frequencies of the PRAMmed variables with REGION are preserved.

[2] For illustrative purposes, we only show this evaluation for the expenditure variables. It can be easily copied for the income variables. The results are similar.

Table 12.9: Univariate frequencies of the PRAMmed variable before and after anonymization

.	0	1	2	3	4	5	6	7	8	9	NA
ROOF		27	1	914	307	711				10	1
ROOF.m		25	1	907	319	712				6	1
TOILET		76	594	817	481					3	
TOILET.m		71	597	816	483					4	
WATER		128	323	304	383	562	197	18	21	35	
WATER.m		134	319	308	378	573	188	16	21	34	
ELECTCON	768	216	8	2							977
ELECTCON.m	761	218	8	3							981
FUELCOOK		1289	21	376	55	36				139	55
FUELCOOK.m		1284	22	383	50	39				143	50
OWNMOTORCYCLE	1883	86									2
OWNMOTORCYCLE.m	1882	86									2
CAR	963	31									977
CAR.m	966	25									
TV	1216	264									491
TV.m	1203	272									496

Table 12.10: Multivariate frequencies of the variables WATER with RURURB before and after anonymization

.	1	2	3	4	5	6	7	8	9
WATER/URB	11	49	270	306	432	183	12	15	21
WATER/RUR	114	274	32	76	130	14	6	6	14
WATER/URB.m	79	220	203	229	402	125	10	12	19
WATER/RUR.m	54	98	105	147	169	63	6	9	15

For conciseness, we restrict ourselves to the analysis of the expenditure variables. The analysis of the income variables can be done in the same way and leads to similar results.

We look at the effect of anonymization on some indicators as discussed in Step 5. Table 12.11 presents the point estimates and bootstrapped confidence interval of the GINI coefficient[3] for the sum of the expenditure components. The calculation of the GINI coefficient and the confidence interval are based on the positive expenditure values. We observe very small changes in the Gini coefficient, that are statistically negligible. We use a visualization to illustrate the impact on utility of the anonymization. Visualizations are discussed in the Section Assessing data utility with the help of data visualizations (in R) and the specific *R* code for this case study is available in the *R* script. The change in the inequality measures is illustrated in Fig. 12.1, which shows the Lorenz curves based on the positive expenditure values before and after anonymization.

Table 12.11: GINI point estimates and bootstrapped confidence intervals for sum of expenditure components

.	before	after
Point estimate	0.510	0.508
Left bound of CI	0.476	0.476
Right bound of CI	0.539	0.538

We compare the mean monthly expenditures (MME) and mean monthly income (MMI) for rural, urban and total

[3] To compute the GINI coefficient, bootstrap to construct the confidence intervals and plot the Lorenz curve we used the *R* packages *laeken, reldist, bootstrap* and *ineq*.

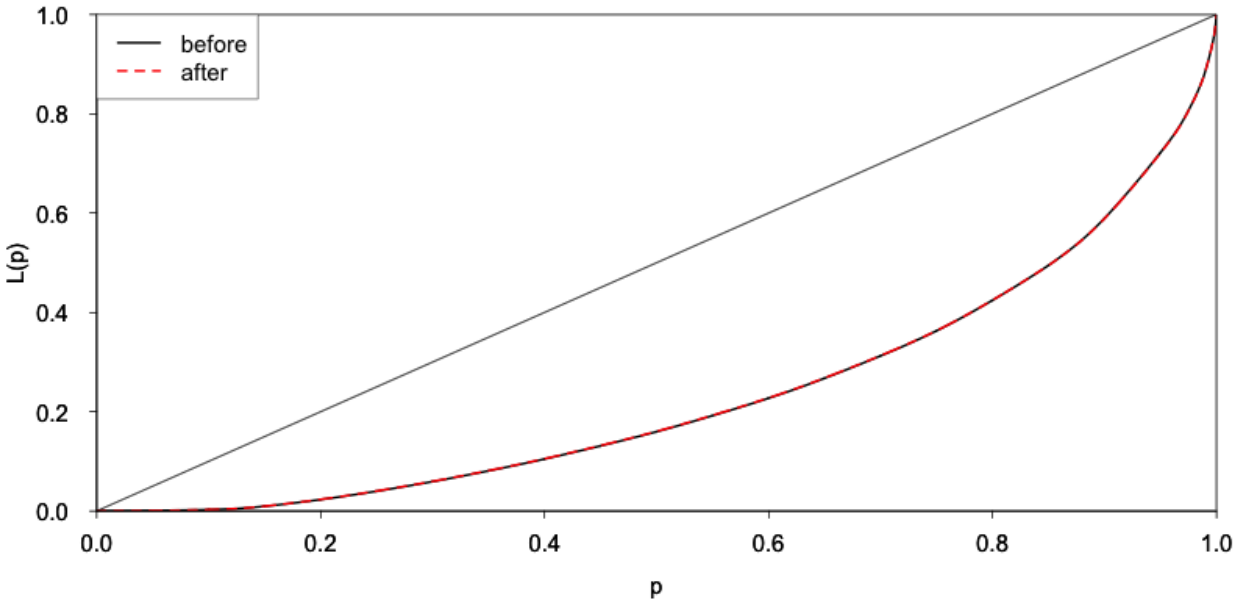

Fig. 12.1: Lorenz curve based on positive total expenditures values

population. The results are shown in Table 12.12. We observe that the chosen levels of noise add only small distortions to the MME and slightly larger changes to the MMI.

Table 12.12: Mean monthly expenditure and mean monthly income per capita by rural/urban

.	before	after
MME rural	400.5	398.5
MME urban	457.3	459.9
MME total	412.6	412.6
MMI rural	397.1	402.2
MMI urban	747.6	767.8
MMI total	472.1	478.5

Table 12.13 shows the share of each of the components of the expenditure variables before and after anonymization.

Table 12.13: Shares of expenditures components

.	TFOODEXP	TALCHEXP	TCLTHEXP	THOUSEXP	TFURNEXP	THLTHEXP
before	0.58	0.01	0.03	0.09	0.02	0.03
after	0.59	0.01	0.03	0.09	0.02	0.03
.	TTRANSEXP	TCOMMEXP	TRECEXP	TEDUEXP	TRESHOTEXP	TMISCEXP
before	0.04	0.02	0.00	0.08	0.03	0.05
after	0.04	0.02	0.00	0.08	0.03	0.05

Anonymization for the creation of a SUF will inevitably lead to some degree of utility loss. It is important to describe this loss in the external report, so that users are aware of the changes in the data. This is described in Step 11 and presented in Appendix C. Appendix C also shows summary statistics and tabulations of the household level variables before and after anonymization.

Merging the household- and individual-level variables

The next step is to merge the treated household variables with the untreated individual variables for the anonymization of the individual level variables. Listing 12.21 shows the steps to merge these files. This also includes the selection of variables used in the anonymization of the individual-level variables. We create the *sdcMicro* object for the anonymization of the individual variables in the same way as for the household variable in Listing 12.9. Subsequently, we repeat Steps 6-10 for the individual-level variables.

Listing 12.21: Merging the files with household and individual-level variables and creating an *sdcMicro* object for the anonymization of the individual-level variables

```
### Select variables (individual level)
# Key variables (individual level)
selectedKeyVarsIND = c('GENDER', 'REL', 'MARITAL', 'AGEYRS',
                       'EDUCY', 'ATSCHOOL', 'INDUSTRY1') # list of selected key variables

# Sample weight (WGTHH, individual weight)
selectedWeightVarIND = c('WGTHH')

# Household ID
selectedHouseholdID = c('IDH')

# No strata

# Recombining anonymized HH datasets and individual level variables
indVars <- c("IDH", "IDP", selectedKeyVarsIND, "WGTHH") # HID and all non HH variables
fileInd <- file[indVars] # subset of file without HHVars

HHmanip <- extractManipData(sdcHH) # manipulated variables HH
HHmanip <- HHmanip[HHmanip[,'IDH'] != 1782,]

fileCombined <- merge(HHmanip, fileInd, by.x= c('IDH'))

fileCombined <- fileCombined[order(fileCombined[,'IDH'],
fileCombined[,'IDP']),]

dim(fileCombined)

# SDC objects with all variables and treated HH vars for
# anonymization of individual level variables
sdcCombined <- createSdcObj(dat = fileCombined, keyVars = selectedKeyVarsIND,
                            weightVar = selectedWeightVarIND, hhId = selectedHouseholdID)
```

12.1.12 Step 6b: Assessing disclosure risk (individual level)

All key variables at the individual level are categorical. Therefore, we can use k-anonymity and the individual and global risk measures (see the Sections Individual risk and Global risk). The hierarchical risk is now of interest, given the household structure in the dataset *fileCombined*, which includes both household- and individual-level variables. The number of individuals (absolute and relative) that violate k-anonymity at the levels 2, 3 and 5 are shown in Table 12.14.

Note: k-anonymity does not consider the household structure and therefore underestimates the risk. Therefore, we are more interested in the individual and global hierarchical risk measures.

Table 12.14: k-anonymity violations

k-anonymity	Number HH violating	Percentage
2	998	9.91%
3	1,384	13.75%
5	2,194	21.79%

The global risk measures can be found using *R* as illustrated in Listing 12.22. The global risk is 0.24%, which corresponds to 24 expected re-identifications. Accounting for the hierarchical structure, this rises to 1.26%, or 127 expected re-identifications. The global risk measures are low compared to the number of k-anonymity violators due to the low sampling weights. The high number of k-anonymity violators is mainly due to the very detailed age variable. The risk measures are based only on the individual level variables, since we assume that the individual and household level variables are be used simultaneously by an intruder. If we would consider an intruder scenario where these variables are used simultaneously by an intruder to re-identify individuals, the household level variables should also be taken into account here. This would results in a high number of key variables.

Listing 12.22: Global risk of the individual-level variables

```
print(sdcCombined, 'risk')
## Risk measures:
##
## Number of observations with higher risk than the main part of the data: 0
## Expected number of re-identifications: 23.98 (0.24 %)
##
## Information on hierarchical risk:
## Expected number of re-identifications: 127.12 (1.26 %)
```

12.1.13 Step 7b: Assessing utility (individual level)

We evaluate the utility measures selected in Step 5 besides some general utility measures. The values computed from the raw data are presented in step 10b to allow for direct comparison with the values computed from the anonymized data.

12.1.14 Step 8b: Choice and application of SDC methods (individual level)

We use the same approach for the anonymization of the individual-level categorical key variables as for the household level categorical variables described earlier: first use global recoding to limit the necessary number of suppressions, then apply local suppressions and finally, if necessary, use of perturbative methods.

The variable AGEYRS (i.e., age in years) has many different values (age in months for children 0 – 1 years and age in years for individuals over 1 year). This level of detail leads to a high level of re-identification risk, given external datasets with exact age as well as knowledge of the exact age of close relatives. We have to reduce the level of detail in the age variables by recoding the age values (see the Section Recoding). First, we recode the values from 15 to 65 in ten-year intervals. Since some indicators related to education are computed from the survey dataset, our first approach is not to recode the age range 0 – 15 years. For children under the age of 1 year, we reduce the level of detail and recode these to 0 years. These recodes are shown in Listing 12.23. We also top-code age at the age of 65 years. This protects individuals with high (rare) age values.

Listing 12.23: Recoding age in 10-year intervals in the range 15 – 65 and top code age over 65 years

```
# Recoding age and top coding age (top code 65), below that 10 year age
# groups, children aged under 1 are recoded 0 (previously in months)
```

(continues on next page)

(continued from previous page)

```

sdcCombined@manipKeyVars$AGEYRS[sdcCombined@manipKeyVars$AGEYRS >= 0 &
sdcCombined@manipKeyVars$AGEYRS < 1] <- 0

sdcCombined@manipKeyVars$AGEYRS[sdcCombined@manipKeyVars$AGEYRS >= 15 &
sdcCombined@manipKeyVars$AGEYRS < 25] <- 20

...

sdcCombined@manipKeyVars$AGEYRS[sdcCombined@manipKeyVars$AGEYRS >= 55 &
sdcCombined@manipKeyVars$AGEYRS < 66] <- 60

# topBotCoding also recalculates risk based on manual recoding above
sdcCombined <- **topBotCoding(obj = sdcCombined, value = 65,
replacement = 65, kind = 'top', column = 'AGEYRS')

table(sdcCombined@manipKeyVars$AGEYRS) # check results
##    0    1    2    3    4    5    6    7    8    9   10   11   12   13   14
##  311  367  340  332  260  334  344  297  344  281  336  297  326  299  263
##   20   30   40   50   60   65
## 1847 1220  889  554  314  325
```

These recodes already reduce the risk to 531 individuals violating 3-anonymity. We could recode the values of age in the lower range according to the age categories users require (e.g., 8 – 11 for education). There are many different categories for different indicators, however, including education indicators. This would reduce the utility of the data for some users. Therefore, we decide to look first at the number of suppressions needed in local suppression after this limited recoding. If the number of suppressions is too high, we can go back and recode age in the range 1 – 14 years.

In Listing 12.24 we demonstrate how one might experiment with local suppression to find the best option. We use local suppression to achieve 3-anonymity (see the Section Local suppression . On the first attempt, we do not specify any importance vector; this leads to many suppressions in the variable AGEYRS (see Table 12.15 below, first row), however. This is undesirable from a utility point of view. Therefore, we decide to specify an importance vector to prevent suppressions in the variable AGEYRS. Suppressing the variable GENDER is also undesirable from the utility point of view. The variable GENDER is a type of variable that should not have suppressions. We set GENDER as variable with the second highest importance. After specifying the importance vector to prevent suppressions of the age variable, there are no age suppressions (see Table 12.15, second row). The total number of suppressions in the other variables increased, however, from 253 to 323 because of the importance vector. This is to be expected because the algorithm without the importance vector minimizes the total number of suppressions by first suppressing values in variables with many categories – in this case, age and gender. Specifying an importance vector prevents reaching this optimality and hence leads to a higher total number of suppressions. There is a trade-off between which variables are suppressed and the total number of suppressions. After specifying an importance vector, the variable REL has many suppressions (see Table 12.15, second row). We choose this second option.

Listing 12.24: Experimenting with different options in local suppression

```
# Copy of sdcMicro object to later undo steps
sdcCopy <- sdcCombined

# Importance vectors for local suppression (depending on utility measures)
impVec1 <- NULL # for optimal suppression
impVec2 <- rep(length(selectedKeyVarsIND), length(selectedKeyVarsIND))
impVec2[match('AGEYRS', selectedKeyVarsIND)] <- 1 # AGEYRS
impVec2[match('GENDER', selectedKeyVarsIND)] <- 2 # GENDER

# Local suppression without importance vector
```

(continues on next page)

(continued from previous page)

```
sdcCombined <- localSuppression(sdcCombined, k = 2, importance = impVec1)

# Number of suppressions per variable
print(sdcCombined, "ls")

## Local Suppression:
##       KeyVar | Suppressions (#) | Suppressions (%)
##       GENDER |                 0 |            0.000
##          REL |                34 |            0.338
##      MARITAL |                 0 |            0.000
##      AGEYRS |               195 |            1.937
##       EDUCY |                 0 |            0.000
##  EDYRSCURRAT |                 3 |            0.030
##     ATSCHOOL |                 0 |            0.000
##    INDUSTRY1 |                21 |            0.209

# Number of suppressions per variable for each value of AGEYRS
table(sdcCopy@manipKeyVars$AGEYRS) - table(sdcCombined@manipKeyVars$AGEYRS)

##  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 20 30 40 50 60 65
##  0  0  0  0  0  0  2  0  2  1  0  1  4  1  5 25 53 37 36 15 13

# Undo local suppression
sdcCombined <- undolast(sdcCombined)

# Local suppression with importance vector on AGEYRS and GENDER
sdcCombined <- localSuppression(sdcCombined, k = 2, importance = impVec2)

# Number of suppressions per variable
print(sdcCombined, "ls")
## Local Suppression:
##       KeyVar | Suppressions (#) | Suppressions (%)
##       GENDER |                 0 |            0.000
##          REL |               323 |            3.208
##      MARITAL |                 0 |            0.000
##       AGEYRS |                 0 |            0.000
##       EDUCY |                 0 |            0.000
##  EDYRSCURRAT |                 0 |            0.000
##     ATSCHOOL |                 0 |            0.000
##    INDUSTRY1 |                 0 |            0.000

# Number of suppressions for each value of the variable AGEYRS
table(sdcCopy@manipKeyVars$AGEYRS) - table(sdcCombined@manipKeyVars$AGEYRS)
##  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 20 30 40 50 60 65
##  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

Table 12.15: Number of suppressions by variable for different variations of local suppression

Local suppression options	GENDER	REL	MARITAL	AGEYRS	EDUCY	EDYRSCURRATT	ATSCHOOL	INDUSTRY1
k = 2, no imp	0	34	0	195	0	3	0	21
k = 2, imp on AGEYRS	0	323	0	0	0	0	0	0

12.1.15 Step 9b: Re-measure risk (individual level)

We re-evaluate the risk measures selected in Step 6b. Table 12.16 shows that local suppression, not surprisingly, has reduced the number of individuals violating 2-anonymity to 0.

Table 12.16: k-anonymity violations

k-anonymity	Number HH violating	Percentage
2	0	0.00 %
3	197	1.96 %
5	518	5.15 %

The hierarchical global risk was reduced to 0.11%, which corresponds to 11.3 expected re-identifications. The highest individual hierarchical re-identification risk is 1.21%. These risk levels would seem acceptable for a SUF.

12.1.16 Step 10b: Re-measure utility (individual level)

We selected two utility measures for the individual variables: primary and secondary education enrollment, both also by gender. These two measures are sensitive to changes in the variables gender (GENDER), age (AGEYRS) and education (EDUCY and EDYRSATCURR), and therefore give a good overview of the impact of the anonymization. As shown in Table 12.17 the anonymization did not change the results. The results of the tabulations in Appendix C confirm these results.

Table 12.17: Net enrollment in primary and secondary education by gender

.	Primary education			Secondary education		
Before	72.6%	74.2%	70.9%	42.0%	44.8%	39.1%
After	72.6%	74.2%	70.9%	42.0%	44.8%	39.1%

12.1.17 Step 11: Audit and reporting

In the audit step, we check whether the data allow for reproduction of published figures from the original dataset and relationships between variables and other data characteristics are preserved in the anonymization process. In short, we check whether the dataset is valid for analytical purposes. There are no figures available that were published from the dataset and need to be reproducible from the anonymized data.

In Step 2, we explored the data characteristics and relationships between variables. These data characteristics and relationships have been mainly preserved, since we took them into account when choosing the appropriate anonymization methods. The variables TANHHEXP and INCTOTGROSSHH are the sums of the individual components, because we added noise to the components and reconstructed the aggregates by summing over the components. Initially, the income variables were all positive. This characteristic has been violated, as a result of noise addition. Since values of the variable AGEYRS were not perturbed, but only recoded and suppressed, we did not introduce unlikely combinations, such as a 60-year-old individual enrolled in primary education. Also, by separating the anonymization process into two parts, one for household-level variables and one for individual-level variables, the values of variables measured at the household level agree for all members of each household.

Furthermore, we drafted two reports, internal and external, on the anonymization of the case study dataset. The internal report includes the methods used, the risk before and after anonymization as well as the reasons for the selected methods and their parameters. The external report focuses on the changes in the data and the loss in utility. Focus here should be on the number of suppressions as well as the perturbative methods (PRAM). This is described in the previous steps.

Note: When creating a SUF, it is inevitable that there will be a loss of information and it is very important for the users to be aware of these changes and release them in a report that accompanies the data.

Appendix C provides examples of an internal and external report of the anonymization process of this dataset. Depending on the users and readers of the reports, the content may differ. The code to this case study shows how to obtain the information for the reports. Some measures are also available in the standard reports generated with the report() function. This is shown in Listing 12.25. The report() function will only use the data available in the *sdcMicro* object, which does not contain all households for sdcHH.

Listing 12.25: Using the report() function for internal and external reports

```
# Create reports with sdcMicro report() function
report(sdcHH, internal = F) # external (brief) report
report(sdcHH, internal = T) # internal (extended) report

# Create reports with sdcMicro report() function
report(sdcCombined, internal = F) # external (brief) report
report(sdcCombined, internal = T) # internal (extended) report
```

12.1.18 Step 12: Data release

The final step is the release of the anonymized dataset together with the external report. Listing 12.26 shows how to collect the data from the *sdcMicro* object with the extractManipData() function. Before releasing the file, we add an individual ID to the file (line number in household). We export the anonymized dataset in as *STATA* file. The Section Read functions in R presents functions for exporting files in other data formats.

Listing 12.26: Exporting the anonymized dataset

```
# Anonymized dataset
# Household variables and individual variables
# extracts all variables, not just the manipulated ones
dataAnon <- extractManipData(sdcCombined, ignoreKeyVars = F, ignorePramVars = F,
                             ignoreNumVars = F, ignoreStrataVar = F)

# Create STATA file
write.dta(dataframe = dataAnon, file= 'Case1DataAnon.dta', convert.dates=TRUE)
```

12.2 Case study 2 - PUF

This case study is a continuation of case study 1 in the Section *Case study 1- SUF* . Case study 1 produces a SUF file. In this case study we use this SUF file to produce a PUF file of the same dataset, which can be freely distributed. The structure of the SUF and PUF releases will be the same. However, the PUF will contain fewer variables and less (detailed) information than the SUF. We refer to the Section *Case study 1- SUF* for a description of the dataset.

Note: It is also possible to directly produce a PUF from a dataset without first creating a SUF.

As in case study 1, we show how the creation of a PUF can be achieved using the open source and free *sdcMicro* package and *R*. A ready-to-run *R* script for this case study and the dataset are also available to reproduce the results

and allow the user to adapt the code (see http://ihsn.org/home/projects/sdc-practice). Extracts of this code are presented in this section to illustrate several steps of the anonymization process.

Note: The choices of methods and parameters in this case study are based on this particular dataset and the results and choices might be different for other datasets.

This case study follows the steps of the SDC process outlined in The SDC Process.

12.2.1 Step 1: Need for disclosure control

The same reasoning as in case study 1 applies: the SUF dataset produced in case study 1 contains data on individuals and households and some variables are confidential and/or sensitive. The decisions made in case study 1 are based on the disclosure scenarios for a SUF release. The anonymization applied for the SUF does not provide sufficient protection for the release as PUF and the SUF file cannot be released as PUF without further treatment. Therefore, we have to repeat the SDC process with a different set of disclosure scenarios based on the characteristics of a PUF release (see Step 4). This leads to different risk measures, lower accepted risk levels and different SDC methods.

12.2.2 Step 2: Data preparation and exploring data characteristics

In order to guarantee consistency between the released PUF and SUF files, which is required to prevent intruders from using the datasets together (SUF users have also access to the PUF file), we have to use the anonymized SUF file to create the PUF file (see also the Section Step 3: Type of release). In this way all information in the PUF file is also contained in the SUF, and the PUF does not provide additional information to an intruder with access to the SUF. We load the required packages to read the data (*foreign* package for *STATA* files) and load the SUF dataset into "file" as illustrated in Listing 12.27. We also load the original data file (raw data) as "fileOrig". We need the raw data to undo perturbative methods used in case study 1 (see Step 8) and to compare data utility measures (see Step 5). To evaluate the utility loss in the PUF, we have to compare the information in the anonymized PUF file with the information in the raw data. For an overview of the data characteristics and a description of the variables in both files, we refer to Step 2 of *Case study 1- SUF* .

Listing 12.27: Loading required packages and datasets

```
# Load required packages
library(foreign)  # for read/write function for STATA
library(sdcMicro) # sdcMicro package

# Set working directory - set to the path on your machine
setwd("/Users/CaseStudy2")

# Specify file name of SUF file from case study 1
fname <- "CaseDataAnon.dta"

# Specify file name of original dataset (raw data)
fnameOrig <- "CaseA.dta"

# Read-in files
file     <- read.dta(fname, convert.factors = TRUE) # SUF file case study 1
fileOrig <- read.dta(fnameOrig, convert.factors = TRUE) # original data
```

We check the number of variables and number of observations of both files and the variable names of the SUF file, as shown in Listing 12.28. The PUF file has fewer records and fewer variables than the original data file, since we removed large households and several variables to generate the SUF file.

Listing 12.28: Number of individuals and variables and variable names

```
# Dimensions of file (observations, variables)
dim(file)
## [1] 10068    49

dim(fileOrig)
## [1] 10574    68

colnames(file) # Variable names
##  [1] "IDH"           "URBRUR"        "REGION"        "HHSIZE"
##  [5] "OWNAGLAND"     "RELIG"         "ROOF"          "TOILET"
##  [9] "WATER"         "ELECTCON"      "FUELCOOK"      "OWNMOTORCYCLE"
## [13] "CAR"           "TV"            "LIVESTOCK"     "LANDSIZEHA"
## [17] "TANHHEXP"      "TFOODEXP"      "TALCHEXP"      "TCLTHEXP"
## [21] "THOUSEXP"      "TFURNEXP"      "THLTHEXP"      "TTRANSEXP"
## [25] "TCOMMEXP"      "TRECEXP"       "TEDUEXP"       "TRESTHOTEXP"
## [29] "TMISCEXP"      "INCTOTGROSSHH" "INCRMT"        "INCWAGE"
## [33] "INCFARMBSN"    "INCNFARMBSN"   "INCRENT"       "INCFIN"
## [37] "INCPENSN"      "INCOTHER"      "WGTPOP"        "IDP"
## [41] "GENDER"        "REL"           "MARITAL"       "AGEYRS"
## [45] "EDUCY"         "EDYRSCURRAT"   "ATSCHOOL"      "INDUSTRY1"
## [49] "WGTHH"
```

To get an overview of the values of the variables, we use tabulations and cross-tabulations for categorical variables and summary statistics for continuous variables. To include the number of missing values ('NA' or other), we use the option useNA = "ifany" in the table() function. For some variables, these tabulations differ from the tabulations of the raw data, due to the anonymization of the SUF file.

In Table 12.18 the variables in the dataset "file" are listed along with concise descriptions of the variables, the level at which they are collected (individual level (IND) or household level (HH)), the measurement type (continuous, semi-continuous or categorical) and value ranges. Note that the dataset contains a selection of 49 variables of the 68 variable selected for the SUF release. The variables have been preselected based on their relevance for data users and some variables were removed while creating a SUF file. The numerical values for many of the categorical variables are codes that refer to values, e.g., in the variable "URBRUR", 1 stands for 'rural' and 2 for 'urban'. More information on the meanings of coded values of the categorical variables is available in the *R* code for this case study.

Any data cleaning, such as recoding missing value codes and removing empty variables, was already done in case study 1. The same holds for removing any direct identifiers. Direct identifiers are not released in the SUF file.

We identified the following sensitive variables in the dataset: variables related to schooling and labor force status as well as income and expenditure related variables. These variables need protection. Whether a variable is considered sensitive may depend on the release type, country and the dataset itself.

Table 12.18: Overview of the variables in the dataset

No.	Variable name	Description	Level	Measurement	Values
1	IDH	Household ID	HH	.	1-2,000
2	IDP	Individual ID	IND	.	1-13
3	REGION	Region	HH	categorical	1-6
4	URBRUR	Area of residence	HH	categorical	1, 2
5	WGTHH	Individual weighting coefficient	HH	weight	31.2-84
6	WGTPOP	Population weighting coefficient	IND	weight	45.8-93
7	HHSIZE	Household size	HH	semi-continuous	1-33
8	GENDER	Gender	IND	categorical	0, 1
9	REL	Relationship to household head	IND	categorical	1-9

Continued on next

Table 12.18 – continued from previous page

No.	Variable name	Description	Level	Measurement	Values
10	MARITAL	Marital status	IND	categorical	1-6
11	AGEYRS	Age in completed years	IND	semi-continuous	0-65
12	RELIG	Religion of household head	HH	categorical	1, 5-7,
13	ATSCHOOL	Currently enrolled in school	IND	categorical	0, 1
14	EDUCY	Highest level of education attended	IND	categorical	1-6
15	EDYRSCUR AT	Years of education for currently enrolled	IND	semi-continuous	1-18
16	INDUSTRY	Industry classification (1-digit)	IND	categorical	1-10
17	ROOF	Main material used for roof	IND	categorical	1-5, 9
18	TOILET	Main toilet facility	HH	categorical	1-4, 9
19	ELECTCON	Electricity	HH	categorical	0-3
20	FUELCOOK	Main cooking fuel	HH	categorical	1-5, 9
21	WATER	Main source of water	HH	categorical	1-9
22	OWNAGLAN	Ownership of agricultural land	HH	categorical	1-3
23	LANDSIZE	Land size owned by household (ha) (agric and non agric)	HH	continuous	0-40
24	OWNMOTORYCLE	Ownership of motorcycle	HH	categorical	0, 1
25	CAR	Ownership of car	HH	categorical	0, 1
26	TV	Ownership of television	HH	categorical	0, 1
27	LIVESTOC	Number of large-sized livestock owned	HH	semi-continuous	0-25
28	INCRMT	Income – Remittances	HH	continuous	
29	INCWAGE	Income - Wages and salaries	HH	continuous	
30	INCFARMBSN	Income - Gross income from household farm businesses	HH	continuous	
31	INCNFARMBSN	Income - Gross income from household nonfarm businesses	HH	continuous	
32	INCRENT	Income - Rent	HH	continuous	
33	INCFIN	Income - Financial	HH	continuous	
34	INCPENSN	Income - Pensions/social assistance	HH	continuous	
35	INCOTHER	Income - Other	HH	continuous	
36	INCTOTGROSHH	Income - Total	HH	continuous	
37	FARMEMP				
38	TFOODEXP	Total expenditure on food	HH	continuous	
39	TALCHEXP	Total expenditure on alcoholic beverages, tobacco and narcotics	HH	continuous	
40	TCLTHEXP	Total expenditure on clothing	HH	continuous	
41	THOUSEXP	Total expenditure on housing	HH	continuous	
42	TFURNEXP	Total expenditure on furnishing	HH	continuous	
43	THLTHEXP	Total expenditure on health	HH	continuous	
43	TTRANSEXP	Total expenditure on transport	HH	continuous	
44	TCOMMEXP	Total expenditure on communication	HH	continuous	
45	TRECEXP	Total expenditure on recreation	HH	continuous	
46	TEDUEXP	Total expenditure on education	HH	continuous	
47	TRESHOTEXP	Total expenditure on restaurants and hotels	HH	continuous	
48	TMISCEXP	Total expenditure on miscellaneous spending	HH	continuous	
49	TANHHEXP	Total annual nominal household expenditures	HH	continuous	

It is always important to ensure that the relationships between variables in the data are preserved during the anonymization process and to explore and take note of these relationships before beginning the anonymization. At the end of the anonymization process before the release of the data, an audit should be conducted, using these initial results, to check that these relationships are maintained in the anonymized dataset (see Step 11).

In our dataset, we identify several relationships between variables that need to be preserved during the anonymization process. The variables "TANHHEXP" and "INCTOTGROSSHH" represent the total annual nominal household expenditure and the total gross annual household income, respectively, and these variables are aggregations of existing income and expenditure components in the dataset.

The variables related to education are available only for individuals in the appropriate age groups and missing for other individuals. In addition, the household-level variables (cf. fourth column of Table 12.18) have the same values for all members in any particular household. The value of household size corresponds to the actual number of individuals belonging to that household in the dataset. As we proceed, we have to take care that these relationships and structures are preserved in the anonymization process.

We assume that the data are collected in a survey that uses simple sampling of households. The data contains two weight coefficients: "WGTHH" and "WGTPOP". The relationship between the weights is $WGTPOP = WGTHH * HHSIZE$. "WGTPOP" is the sampling weight for the households and "WGTHH" is the sampling weight for the individuals to be used for disclosure risk calculations. "WGTHH" is used for computing individual-level indicators (such as education) and "WGTPOP" is used for population level indicators (such as income indicators). There are no strata variables available in the data. We will use "WGTPOP" for the anonymization of the household variables and "WGTHH" for the anonymization of the individual-level variables.

12.2.3 Step 3: Type of release

In this case study, we assume that the file will be released as a PUF, which will be freely available to anyone interested in the data (see the Section Conditions for PUFs for the conditions and more information on the release of PUFs). The PUF release is intended for users with lower information requirements (e.g., students) and researchers interested in the structure of the data and willing to do preliminary research. The PUF file can give an idea to the researcher whether it is worthwhile for their research to apply for access to the SUF file. Researchers willing to do more in-depth research will most likely apply for SUF access. Generally, users of a PUF file are not restricted by an agreement that prevents them from using the data to re-identify individuals and hence the accepted risk level is much lower than in the case of the SUF and the set of released variables is limited.

12.2.4 Step 4: Intruder scenarios and choice of key variables

Next, based on the release type, we reformulate the intruder scenarios for the PUF release. This leads to the selection of a set of quasi-identifiers. Since this case study is based on a demo dataset, we do not have a real context and we cannot define exact disclosure scenarios. Therefore, we make hypothetical assumptions on possible disclosure scenarios. We consider two types of disclosure scenarios: 1) matching with other publically available datasets and 2) spontaneous recognition. Since the dataset will be distributed as PUF, there are de facto no restrictions on the use of the dataset by intruders.

For the sake of illustration, we assume that population registers are available with the demographic variables gender, age, place of residence (region, urban/rural), religion and other variables such as marital status and variables relating to education and professional status that are also present in our dataset. In addition, we assume that there is a publically available cadastral register on land ownership. Based on this analysis of available data sources, we have selected in case study 1 the variables "REGION", "URBRUR", "HHSIZE", "OWNAGLAND", "RELIG", "GENDER", "REL" (relationship to household head), "MARITAL" (marital status), "AGEYRS", "INDUSTRY1" and two variables relating to school attendance as categorical quasi-identifiers, the expenditure and income variables as well as LANDSIZEHA as continuous quasi-identifiers. According to our assessment, these variables might enable an intruder to re-identify an individual or household in the dataset by matching with other available datasets. The key variables for PUF release generally coincide with the key variables for the SUF release. Possibly, more variables could be added, since the user has more possibilities to match the data extensively and is not bound by any contract, as is in the case of the SUF file. Equally, some key variables in the SUF file may not be released in the PUF file and, as a consequence, these variables are removed from the list of key variables.

Upon further consideration, this initial set of identifying variables is too large for a PUF release, as the number of possible combinations (keys) is very high and hence many respondents could be identified based on these variables. Therefore, we decide to limit the set of key variables, by excluding variables from the dataset for PUF release. The choice of variables to be removed is led by the needs of the intended PUF users. Assuming the typical users are mainly interested in aggregate income and expenditure data, we can therefore remove from the initial set of key

variables “OWNAGLAND”, “RELIG” and “LANDSIZEHA” at the household level and “EDYRSCURRAT” and “ATSCHOOL” at the individual level.

Note: These variables will not be released in the PUF file.

We also remove the income and expenditure components from the list of key variables, since we reduce their information content by building proportions (see Step 8a). The list of the remaining key variables is presented in Table 12.19.

Table 12.19: Overview of selected key variables for PUF file

Variable name	Variable description	Measurement level
REGION	region	Household, categorical
URBRUR	area of residence	Household, categorical
HHSIZE	household size	Household, categorical
TANHHEXP	total expenditure	Household, continuous
INCTOTGROSSHH	total income	Household, continuous
GENDER	gender	Individual, categorical
REL	relationship to household head	Individual, categorical
MARITAL	marital status	Individual, categorical
AGEYRS	age in completed years	Individual, semi-continuous/categorical
EDUCY	highest level of education completed	Individual, categorical
INDUSTRY1	industry classification	Individual, categorical

The decision to release the dataset as a PUF means the level of anonymization will be relatively high and consequently, the variables are less detailed (e.g., after recoding) and a scenario of spontaneous recognition is less likely. Nevertheless, we should check for rare combinations or unusual patterns in the variables. Variables that may lead to spontaneous recognition in our sample are amongst others “HHSIZE” (household size) as well as “INCTOTGROSSHH” (aggregate income) and “TANHHEXP” (total expenditure). Large households and households with high income are easily identifiable, especially when combined with other identifying variables such as a geographical identifier (“REGION”). There might be only one or a few households/individuals in a certain region with a high income, such as the local doctor. Variables that are easily observable and known by neighbors such as “ROOF”, “TOILET”, “WATER”, “ELECTCON”, “FUELCOOK”, “OWNMOTORCYCLE”, “CAR”, “TV” and “LIVESTOCK” may also need protection depending on what stands out in the community, since a user might be able to identify persons (s)he knows. This is called the nosy-neighbor scenario.

12.2.5 Step 5: Data key uses and selection of utility measures

A PUF file contains less information and the file is generally used by students as a teaching file, by researchers to get an idea about the data structure, and for simple analyses. The users have generally lower requirements than for a SUF file and the results of analysis may be less precise. The researcher interested in a more detailed dataset, would have to apply for access to the SUF file. Therefore, we select more aggregate utility measures for the PUF file that reflect the intended use of a PUF file. Data intensive measures, such as the Gini coefficient, cannot be computed from the PUF file. Besides the standard utility measures, such as tabulations, we evaluate the decile dispersion ratio and a regression with the income deciles as regressand.

To measure the information loss, we should compare the initial data file before any anonymization (including the anonymization for the SUF) with the file after the anonymization for the PUF. Comparing the files directly before and after the PUF anonymization would underestimate the information loss, as this would omit the information loss due to SUF anonymization. Therefore, in Step 2, we also loaded the raw dataset.

Hierarchical (household) structure

As noted in case study 1, the data has a household structure. For the SUF release, we protected large households by removing these from the dataset. Since some variables are measured on the household level and thus have identical values for each household member, the values of the household variables should be treated in the same way for each household member (see the Section Anonymization of the quasi-identifier household size). Therefore, we first anonymize only the household variables. After this, we merge them with the individual-level variables and then anonymize the individual-level and household-level variables jointly.

Since the data has a hierarchical structure, Steps 6 through 10 are repeated twice: Steps 6a through 10a are for the household-level variables and Steps 6b through 10b for the combined dataset. In this way, we ensure that household-level variable values remain consistent across household members for each household and the household structure cannot be used to re-identify individuals. This is further explained in the Sections Levels of risk and Household structure.

Before continuing to Step 6a, we select the categorical key variables, continuous key variables and any variables selected for use in PRAM routines, as well as household-level sampling weights in *R*. We also collect the variable names of the variables that will not be released. The PRAM variables are variables select for the PRAM routine, which we discuss further in Step 8a. We extract these selected household variables from the SUF dataset and save them as "fileHH". The choice of PRAM variables is further explained in Step 8a. Listing 12.29 illustrates how these steps are done in *R* (see also the Section Household structure).

Listing 12.29: Selecting the variables for the household-level anonymization

```
# Categorical key variables at household level
selectedKeyVarsHH <- c('URBRUR', 'REGION', 'HHSIZE')

# Continuous key variables
numVarsHH <- c('TANHHEXP', 'INCTOTGROSSHH')

# PRAM variables
pramVarsHH <- c('ROOF', 'TOILET', 'WATER', 'ELECTCON',
                'FUELCOOK', 'OWNMOTORCYCLE', 'CAR', 'TV', 'LIVESTOCK')

# Household weight
weightVarHH <- c('WGTPOP')

# Variables not suitable for release in PUF (HH level)
varsNotToBeReleasedHH <- c("OWNAGLAND", "RELIG", "LANDSIZEHA")

# Vector with names of all HH level variables
HHVars <- c('IDH', selectedKeyVarsHH, pramVarsHH, numVarsHH, weightVarHH)

# Create subset of file with only HH level variables
fileHH <- file[,HHVars]
```

Every household has the same number of entries as it has members (e.g., a household of three will be repeated three times in "fileHH"). Before analyzing the household-level variables, we select only one entry per household, as illustrated in Listing 12.30. This is further explained in the Section Household structure. In the same way we extract "fileOrigHH" from "fileOrig". "fileOrigHH" contains all variables from the raw data, but contains every household only once. We need "fileOrigHH" in Steps 8a and 10a for undoing some perturbative methods used in the SUF file and computing utility measures from the raw data respectively.

Listing 12.30: Taking a subset with only households

```
# Remove duplicated rows based on IDH, one row per household in fileHH
fileHH <- fileHH[which(!duplicated(fileHH$IDH)),] # SUF file
fileOrigHH <- fileOrig[which(!duplicated(fileOrig$IDH)),] # original dataset
```

(continues on next page)

(continued from previous page)

```
# Dimensions of fileHH
dim(fileHH)
## [1] 1970   16

dim(fileOrigHH)
## [1] 2000   68
```

The file "fileHH" contains 1,970 households and 16 variables. We are now ready to create our *sdcMicro* object with the corresponding variables we selected in Listing 12.28. For our case study, we will create an *sdcMicro* object called "sdcHH" based on the data in "fileHH", which we will use for steps 6a – 10a (see Listing 12.34).

Note: When the sdcMicro object is created, the sdcMicro package automatically calculates and stores the risk measures for the data.

This leads us to Step 6a.

Listing 12.31: Creating a *sdcMicro* object for the household variables

```
# Create initial sdcMicro object for household level variables
sdcHH <- createSdcObj(dat = fileHH, keyVars = selectedKeyVarsHH,
                      pramVars = pramVarsHH, weightVar = weightVarHH, numVars =␣
↪numVarsHH)
numHH <- length(fileHH[,1]) # number of households
```

12.2.6 Step 6a: Assessing disclosure risk (household level)

Based on the key variables selected in the disclosure scenarios, we can evaluate the risk at the household level. The PUF risk measures show a lower risk level than in the SUF file after anonymization in case study 1. The reason is that the set of key variables is smaller, since some variables will not be released in the PUF file. Removing (key) variables reduces the risk, and it is one of the most straightforward SDC methods.

As a first measure, we evaluate the number of households violating k-anonymity at the levels 2, 3 and 5. Table 12.20 shows the number of violating households as well as the percentage of the total number of households. Listing 12.32 illustrates how to find these values with *sdcMicro*. The print() function in *sdcMicro* shows only the values for thresholds 2 and 3. Values for other thresholds can be calculated manually by summing up the frequencies smaller than the k-anonymity threshold, as shown in Listing 12.32. The number of violators is already at a low level, due to the prior anonymization of the SUF file and the reduced set of key variables.

Table 12.20: Number and proportion of households violating k-anonymity

k-anonymity level	Number of HH violating	Percentage of total number of HH
2	0	0.0%
3	18	0.9%
5	92	4.7%

Listing 12.32: Showing number of households violating k-anonymity for levels 2, 3 and 5

```
# Number of observations violating k-anonymity (thresholds 2 and 3)
print(sdcHH)
```

(continues on next page)

(continued from previous page)

```
## Infos on 2/3-Anonymity:
##
## Number of observations violating
##   - 2-anonymity: 0
##   - 3-anonymity: 18
##
## Percentage of observations violating
##   - 2-anonymity: 0.000 %
##   - 3-anonymity: 0.914 %
------------------------------------------------------------------------

# Calculate sample frequencies and count number of obs. violating k(5) - anonymity
kAnon5 <- sum(sdcHH@risk$individual[,2] < 5)
kAnon5
## [1] 92

# As percentage of total
kAnon5 / numHH
## [1] 0.04670051
```

It is often useful to view the records of the household(s) that violate k-anonymity. This might help to find which variables cause the uniqueness of these households; this can then be used later when choosing appropriate SDC methods. Listing 12.32 shows how to access the values of the households violating 3 and 5-anonymity. Not surprisingly, the variable "HHSIZE" is responsible for many of the unique combinations and the origin of much of the risk. This is even the case after removing large households for the SUF release.

Listing 12.33: Showing records of households that violate k-anonymity

```
# Show values of key variable of records that violate k-anonymity
fileHH[sdcHH@risk$individual[,2] < 3, selectedKeyVarsHH] # for 3-anonymity
fileHH[sdcHH@risk$individual[,2] < 5, selectedKeyVarsHH] # for 5-anonymity
```

We also assess the disclosure risk of the categorical variables with the individual and global risk measures as described in the Sections Individual risk and Global risk. In "fileHH" every entry represents a household. Therefore, we use the individual non-hierarchical risk here, where the individual refers in this case to a household. "fileHH" is a subset of the complete dataset and contains only households and has, contrary to the complete dataset, no hierarchical structure. In Step 6b, we evaluate the hierarchical risk in the dataset "file", the dataset containing both households and individuals. The individual and global risk measures automatically take into consideration the household weights, which we defined in Listing 12.29. In our file, the global risk measure calculated using the chosen key variables is lower than 0.01% (the smallest reported value is 0.01%, in fact the global risk is 0.0000642 %). This percentage is extremely low and corresponds to 0.13 expected re-identifications. The results are also shown in Listing 12.34. This low figure can be explained by the relatively small sample size of 0.25% of the total population (see case study 1). Furthermore, one should keep in mind that this risk measure is based only on the categorical quasi-identifiers at the household level.

Listing 12.34: Printing global risk measures

```
print(sdcHH, "risk")
## Risk measures:
##
## Number of observations with higher risk than the main part of the data: 0
## Expected number of re-identifications: 0.13 (0.01 %)
```

The global risk measure does not provide information about the spread of the individual risk measures. There might be a few households with relatively high risk, while the global (average) risk is low. Therefore we check the highest individual risk as shown in Listing 12.35. The individual risk of the household with the highest risk is 0.1 %, which is still very low.

Listing 12.35: Determining the highest individual risk

```
# Highest individual risk
max(sdcHH@risk$individual[, "risk"])
## [1] 0.001011633
```

Since the selected key variables at the household level are both categorical and numerical, the individual and global risk measures based on frequency counts do not completely reflect the disclosure risk of the entire dataset. When generating the SUF file, we concluded that recoding of continuous variables to make them all categorical would likely not satisfy the needs of the SUF users. For the PUF file it is acceptable to recode continuous variables, such as income and expenditures since PUF content is typically less detailed. In Step 8a we will recode these variables into deciles and convert them into categorical variables. Therefore, we exclude these variables from the risk calculations now We take these variables into account while remeasuring risk after anonymization.

12.2.7 Step 7a: Assessing utility measures (household level)

The utility of the data does not only depend on the household level variables, but on the combination of household-level and individual-level variables. Therefore, it is not useful to evaluate all the utility measures selected in Step 5 at this stage, i.e., before anonymizing the individual level variables. We restrict the initial measurement of utility to those measures that are solely based on the household variables. In our dataset, these are the measures related to income and expenditure and their distributions. The results are presented in Step 10a, together with the results after anonymization, which allow direct comparison. If after the next anonymization step it appears that the data utility has been significantly decreased by the suppression of some household level variables, we can return to this step.

Note: To analyze the utility loss, the utility measures before anonymization have to be calculated from the raw data and not from the anonymized SUF file.

Not all measures from case study 1 can be computed from the PUF file, since the information content is lower. The set of utility measures we use to evaluate the information loss in the PUF file consists of measures that need less detailed variables. This reflects the lower requirements a PUF user has on the dataset.

12.2.8 Step 8a: Choice and application of SDC methods (household level)

This step is divided into the anonymization of the categorical key variables and the continuous key variables, since different methods are used for both sets of variables. As already discussed in Step 4, we do not release all variables in the PUF file. At the household level "RELIG" (religion of household head), "OWNAGLAND" (land ownership) and "LANDSIZEHA" (plot size in ha) are not released in addition to the variables removed for the SUF release. For the sake of illustration, we assume that the variable "RELIG" is too sensitive and the variables "OWNAGLAND" and "LANDSIZEHA" are too identifying.

Categorical variables

We are now ready to move on to the choice of SDC methods for the categorical variables on the household level in our dataset. In the SUF file we already recoded some of the key variables and used local suppression. We only have three categorical key variables at the household level; "URBRUR", "REGION" and "HHSIZE". The selected categorical key variables at the household level are not suitable for recoding at this point, since the values cannot be grouped further. "URBRUR" has only two distinct categories and "REGION" has only six non-combinable regions. As noted before, the variable "HHSIZE" can be reconstructed by a headcount per household. Therefore, recoding of this variable alone does not lead to disclosure control.

Due to the relatively low risk of re-identification based on the three selected categorical household level variables, it is possible in this case to use an option like local suppression to achieve our desired level of risk. Applying local

suppression when initial risk is relatively low will likely only lead to suppression of few observations and thus limit the loss of utility. If, however, the data had been measured to have a relatively high risk, then applying local suppression without previous recoding would likely result in a large number of suppressions and greater information loss. Efforts such a recoding should be taken first before suppressing values in cases where risk is initially measured as high. Recoding will reduce risk with little information loss and thus the number of suppressions, if local suppression is applied as an additional step.

We apply local suppression to reach 5-anonymity. The chosen level of five is higher than in the SUF release and is based on the release type as PUF. This leads to a total of 39 suppressions, all in the variable "HHSIZE". As explained earlier, suppression of the value of the variable "HHSIZE" does not lead to actual suppression of this information. Therefore, we redo the local suppression, but this time we tell *sdcMicro* to, if possible, not suppress "HHSIZE" but one of the other variables. Alternatively, we could remove households with suppressed values of the variable "HHSIZE", remove large households or split households.

In *sdcMicro* it is possible to tell the algorithm which variables are important and less important for making small changes (see also the Section Local suppression). To prevent values of the variable "HHSIZE" being suppressed, we set the importance of "HHSIZE" in the importance vectors to the highest (i.e., 1). We try two different importance vectors: the first where "REGION" is more important than "URBRUR" and the second with the importance of "REGION" and "URBRUR" swapped. Listing 12.36 shows how to apply local suppression and put importance on the variable "HHSIZE".

Note: In Listing 12.36 we use the undolast() function in sdcMicro to go one step back after we had first applied local suppression with no importance vector.

The undolast() function restores the *sdcMicro* object back to the previous state (i.e., before we applied local suppression), which allows us to rerun the same command, but this time with an importance vector set. The undolast() function can only be used to go one step back.

The suppression patterns of the three different options are shown in Table 12.21. The importance is clearly reflected in the number of suppressions per variable. The total number of suppressions is with an importance vector higher than without an importance vector (44/73 vs. 39), but 5-anonymity is achieved in the dataset with no suppressions in the variable "HHSIZE". This means that we do not have to remove or split households. The variable "REGION" is the type of variable that should not have any suppressions either. From that perspective we chose the third option. This leads to more suppressions, but no suppressions in "HHSIZE" and as few as possible in "REGION".

Table 12.21: Number of suppressions by variable after local suppression with and without importance vector

Key variable	Number of suppressions and proportion of total		
.	No importance vector	Importance HHSIZE, URBRUR, REGION	Importance HHSIZE, REGION, URBRUR
URBRUR	0 (0.0 %)	2 (0.1 %)	61 (3.1 %)
REGION	0 (0.0 %)	42 (2.1 %)	12 (0.6 %)
HHSIZE	39 (2.0 %)	0 (0.0 %)	0 (0.0 %)

Listing 12.36: Local suppression with and without importance vector

```
# Local suppression to achieve 5-anonimity
sdcHH <- localSuppression(sdcHH, k = 5, importance = NULL) # no importance vector
print(sdcHH, "ls")
## Local Suppression:
##  KeyVar | Suppressions (#) | Suppressions (%)
##  URBRUR |                0 |            0.000
```

(continues on next page)

(continued from previous page)

```
##  REGION |                0 |            0.000
##  HHSIZE |               39 |            1.980
## ------------------------------------------------------------------------

# Undo suppressions to see the effect of an importance vector
sdcHH <- undolast(sdcHH)

# Redo local suppression minimizing the number of suppressions in HHSIZE
sdcHH <- localSuppression(sdcHH, k = 5, importance = c(2, 3, 1))

print(sdcHH, "ls")
## Local Suppression:
##  KeyVar | Suppressions (#) | Suppressions (%)
##  URBRUR |                2 |            0.102
##  REGION |               42 |            2.132
##  HHSIZE |                0 |            0.000
## ------------------------------------------------------------------------

# Undo suppressions to see the effect of a different importance vector
sdcHH <- undolast(sdcHH)

# Redo local suppression minimizing the number of suppressions in HHSIZE
sdcHH <- localSuppression*(sdcHH, k = 5, importance = c(3, 2, 1))

print(sdcHH, "ls")
## Local Suppression:
##  KeyVar | Suppressions (#) | Suppressions (%)
##  URBRUR |               61 |            3.096
##  REGION |               12 |            0.609
##  HHSIZE |                0 |            0.000
## ------------------------------------------------------------------------
```

In case study 1 we applied invariant PRAM to the variables "ROOF", "TOILET", "WATER", "ELECTCON", "FUELCOOK", "OWNMOTORCYCLE", "CAR", "TV" and "LIVESTOCK", since these variables are not sensitive and were not selected as quasi-identifiers because we assumed that there are no external data sources containing this information that could be used for matching. Values can be easily observed or be known to neighbors, however, and therefore are important, together with other variables, for the nosy neighbor scenario. For the PUF release we would like to level of uncertainty by increasing the number of changes. Therefore, we redo PRAM with a different transition matrix. As discussed in the Section PRAM (Post RAndomization Method), the invariant PRAM method has the property that the univariate distributions do not change. To maintain this property, we reapply PRAM to the raw data, rather than to the already PRAMmed variables in the SUF file.

Listing 12.37 illustrates how to apply PRAM. We use the original values to apply PRAM and replace the values in the *sdcMicro* object with these values. We choose the parameter 'pd', the lower bound for the probability that a value is not changed, to be relatively low at 0.6. This is a lower value than the 0.8 used in the SUF file and will lead to a higher number of changes (cf. Listing 12.17). This is acceptable for a PUF file and introduces more uncertainty as required for a PUF release. Listing 12.37 also shows the number of changed records per variables. Because PRAM is a probabilistic method, we set a seed for the random number generator before applying PRAM to ensure reproducibility of the results.

Note: In some cases the choice of the seed matters. The choice of seed changes the results.

The seed should not be released, since it allows for reconstructing the original values if combined with the transition matrix. The transition matrix can be released: this allows for consistent statistical inference by correcting the statistical methods used if the researcher has knowledge about the PRAM method (at this point *sdcMicro* does not allow to

retrieve the transition matrix).

Listing 12.37: Applying PRAM

```
# PRAM
set.seed(10987)

# Replace PRAM variables in sdcMicro object sdcHH with the original raw values
sdcHH@origData[,pramVarsHH] <- fileHH[match(fileHH$IDH, fileOrigHH$IDH), pramVarsHH]
sdcHH@manipPramVars <- fileHH[match(fileHH$IDH, fileOrigHH$IDH), pramVarsHH]

sdcHH <- pram(obj = sdcHH, pd = 0.6)
## Number of changed observations:
## - - - - - - - - - - -
## ROOF != ROOF_pram : 305 (15.48%)
## TOILET != TOILET_pram : 260 (13.2%)
## WATER != WATER_pram : 293 (14.87%)
## ELECTCON != ELECTCON_pram : 210 (10.66%)
## FUELCOOK != FUELCOOK_pram : 315 (15.99%)
## OWNMOTORCYCLE != OWNMOTORCYCLE_pram : 95 (4.82%)
## CAR != CAR_pram : 255 (12.94%)
## TV != TV_pram : 275 (13.96%)
## LIVESTOCK != LIVESTOCK_pram : 109 (5.53%)
```

PRAM has changed values within the variables according to the invariant transition matrices. Since we used the invariant PRAM method (see the Section PRAM (Post RAndomization Method)), the absolute univariate frequencies remain approximately unchanged. This is not the case for the multivariate frequencies. In Step 10a we compare the multivariate frequencies before and after anonymization for the PRAMmed variables.

Continuous variables

We have selected the variables "INCTOTGROSSHH" (total income) and "TANHHEXP" (total expenditure) as numerical quasi-identifiers, as discussed in Step 4. In Step 5 we identified variables having high interest for the users of our data: many users use the data for measuring inequality and expenditure patterns. The noise addition in the SUF file does not protect these variables sufficiently, especially, because outliers are not protected. Therefore, we decide to recode total income and total expenditure into deciles.

As with PRAM, we want to compute the deciles from the raw data rather than from the perturbed values in the SUF file. We compute the deciles directly from the raw data and overwrite these values in the *sdcMicro* object. Subsequently, we compute the mean of each decile from the raw data and replace the values for total income and total expenditures with the mean of the respective decile. In this way the mean of both variables does not change. This approach can be interpreted as univariate microaggregation with very large groups (group size n/10) with the mean as replacement value (see the Section Microaggregation).

The information in the income and expenditure variables by component is too sensitive to release as PUF, and, summing those variables would allow an intruder to reconstruct the totals. PUF users might however be interested in the shares. Therefore, we decide to keep the income and expenditure components as proportions of the raw totals, rounded to two digits. The anonymization of the income and expenditure variables is shown in Listing 12.38.

Listing 12.38: Anonymization of income and expenditure variables

```
# Create bands (deciles) for income and expenditure variables
(aggregates) based on the original data
decExp <- as.numeric(cut(fileOrigHH[match(fileHH$IDH, fileOrigHH$IDH), "TANHHEXP"],
                         quantile(fileOrigHH[match(fileHH$IDH, fileOrigHH$IDH),
"TANHHEXP"],
                                  (0:10)/10, na.rm = T),
                         include.lowest = TRUE, labels = c(1, 2, 3, 4, 5, 6, 7, 8, 9,␣
↪10)))
```

(continues on next page)

(continued from previous page)

```
 decInc <- as.numeric(cut(fileOrigHH[match(fileHH$IDH, fileOrigHH$IDH), "INCTOTGROSSHH"],
                          quantile(fileOrigHH[match(fileHH$IDH, fileOrigHH$IDH), "INCTOTGROSSHH"],
                                   (0:10)/10, na.rm = T),
                          include.lowest = TRUE, labels  = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)))

# Mean values of deciles
decExpMean <- round(sapply(split(fileOrigHH[match(fileHH$IDH, fileOrigHH$IDH), "TANHHEXP"],
                                 decExp), mean))
decIncMean <- round(sapply(split(fileOrigHH[match(fileHH$IDH, fileOrigHH$IDH), "INCTOTGROSSHH"],
                                 decInc), mean))

# Replace with mean value of decile
sdcHH@manipNumVars$TANHHEXP <- decExpMean[match(decExp,names(decExpMean))]
sdcHH@manipNumVars$INCTOTGROSSHH <- decIncMean[\ match(decInc, names(decIncMean))]

# Recalculate risks after manually changing values in sdcMicro object calcRisks(sdcHH)
calcRisks(sdcHH)

# Extract data from sdcHH
HHmanip <- extractManipData(sdcHH) # manipulated variables HH

# Keep components of expenditure and income as share of total,
# use original data since previous data was perturbed
compExp <- c('TFOODEXP', 'TALCHEXP', 'TCLTHEXP', 'THOUSEXP',
             'TFURNEXP', 'THLTHEXP', 'TTRANSEXP', 'TCOMMEXP',
             'TRECEXP', 'TEDUEXP', 'TRESTHOTEXP', 'TMISCEXP')
compInc <- c('INCRMT',  'INCWAGE', 'INCFARMBSN', 'INCNFARMBSN',
             'INCRENT', 'INCFIN',  'INCPENSN',   'INCOTHER')
HHmanip <- cbind(HHmanip, round(fileOrigHH[match(fileHH$IDH, fileOrigHH$IDH), compExp] /
                                fileOrigHH[match(fileHH$IDH, fileOrigHH$IDH),
                                           "TANHHEXP"], 2))
HHmanip <- cbind(HHmanip, round(fileOrigHH[match(fileHH$IDH, fileOrigHH$IDH), compInc] /
                                fileOrigHH[match(fileHH$IDH, fileOrigHH$IDH),
                                           "INCTOTGROSSHH"], 2))
```

12.2.9 Step 9a: Re-measure risk (household level)

For the categorical variables, we conclude that we have achieved 5-anonymity in the data with local suppression. 5-anonymity also implies 2- and 3-anonymity. The global risk stayed close to zero (as the expected number of re-identifications), which is very low. Therefore, we conclude that based on the categorical variables, the data has been sufficiently anonymized. One should keep in mind that the anonymization methods applied are complementing the ones used for the SUF.

Note: The methods selected methods in this case study alone would not be sufficient to protect the data set for a PUF release.

We have reduced the risk of spontaneous recognition of households, by removing the variable “LANDSIZEHA” and

PRAMming the variables identified to be important in the nosy neighbor scenario. An intruder cannot know with certainty whether a household that (s)he recognizes in the data is the correct household, due to the noise in these variables.

These measures refer only to the categorical variables. To evaluate the risk of the continuous variables we could use an interval measure or closest neighbor algorithm. These risk measures are discussed in the Section Risk measures for continuous variables. We chose to use an interval measure, since exact value matching is not our largest concern based on the assumed scenarios and external data sources. Instead, datasets with similar values but not the exact same values could be used for matching. Here the main concern is that the values are sufficiently far from the original values, which is measured with an interval measure.

Listing 12.39 shows how to evaluate the interval measure for the variables "INCTOTGROSSHH" and "TANHHEXP" (total income and expenditure). The different values of the parameter k in the function dRisk() define the size of the interval around the original value as a function of the standard deviation, as explained in the Section Interval measure . The larger k, the larger the intervals, the higher the probability that a perturbed value is in the interval around the original value and the higher the risk measure. The results are satisfactory, especially when keeping in mind that there are only 10 distinct values in the dataset (the means of each of the deciles). All outliers have been recoded. Looking at the proportions of the components, we do not detect any outliers (households with an unusual high or low spending pattern in one component).

Listing 12.39: Measuring risk of re-identification of continuous variables

```
# Risk evaluation continuous variables
dRisk(sdcHH@origData[,c("TANHHEXP", "INCTOTGROSSHH")],
      xm = sdcHH@manipNumVars[,c("TANHHEXP", "INCTOTGROSSHH")], k = 0.01)
## [1] 0.4619289

dRisk(sdcHH@origData[,c("TANHHEXP", "INCTOTGROSSHH")],
      xm = sdcHH@manipNumVars[,c("TANHHEXP", "INCTOTGROSSHH")], k = 0.02)
## [1] 0.642132

dRisk(sdcHH@origData[,c("TANHHEXP", "INCTOTGROSSHH")],
      xm = sdcHH@manipNumVars[,c("TANHHEXP", "INCTOTGROSSHH")], k = 0.05)
## [1] 0.8258883
```

12.2.10 Step 10a Re-measure utility (household level)

The utility in the data has decreased compared to the raw data, mainly because variables were completely removed. Many of the utility measures used in case study 1 are not applicable to the PUF file. However, by replacing the deciles with their means, we can still use the income and expenditure variables for arithmetic operations. Also the shares of the income and expenditure components can still be used, since they are based on the raw data.

We select two additional utility measures: the decile dispersion ratio and the share of total consumption by the poorest decile. The decile dispersion ratio is the ratio of the average income of the top decile and the average income of the bottom decile. Listing 12.40 shows how to compute these from the raw data and the household variables after anonymization. Table 12.22 presents the estimated values. The differences are small and mainly due to the removed households.

Table 12.22: Comparison of utility measures

.	Raw data	PUF file
Decile dispersion ratio	24.12	23.54
Share of consumption by the poorest decile	0.0034	0.0035

Listing 12.40: Computation of decile dispersion ratio and share of total consumption by the poorest decile

```
# Decile dispersion ratio
# raw data
mean(tail(sort(fileOrigHH$INCTOTGROSSHH), n = 200)) /
  mean(head(sort(fileOrigHH$INCTOTGROSSHH), n = 200))
## [1] 24.12152

mean(tail(sort(HHmanip$INCTOTGROSSHH), n = 197)) /
  mean(head(sort(HHmanip$INCTOTGROSSHH), n = 197))
## [1] 23.54179

# Share of total consumption by the poorest decile households
sum(head(sort(fileOrigHH$TANHHEXP), n = 200)) / sum(fileOrigHH$TANHHEXP)
## [1] 0.003411664

sum(head(sort(HHmanip$TANHHEXP), n = 197)) / sum(HHmanip$TANHHEXP)
## [1] 0.003530457
```

Merging the household- and individual-level variables

The next step is to merge the treated household variables with the untreated individual variables for the anonymization of the individual level variables. Listing 12.41 shows the steps to merge these files. This also includes the selection of variables used in the anonymization of the individual-level variables. We create the *sdcMicro* object for the anonymization of the individual variables in the same way as for the household variable in Listing 12.31. Generally, at this stage, the household level and individual level variables should be combined and quasi-identifiers at both levels be used (see the Section Levels of risk). Unfortunately, in our dataset, this leads to long computation times. Therefore, we create two *sdcMicro* objects, one with all key variables ("sdcCombinedAll") and one with only the individual level key variables ("sdcCombined"). In Step 6b we compare the risk measures for both cases and in Step 8b we discuss alternative approaches to keeping the complete set of variables. We now repeat Steps 6-10 for the individual-level variables.

Listing 12.41: Merging the files with household and individual-level variables and creating an *sdcMicro* object for the anonymization of the individual-level variables

```
### Select variables (individual level)
selectedKeyVarsIND = c('GENDER', 'REL', 'MARITAL',
                       'AGEYRS', 'EDUCY', 'INDUSTRY1') # list of selected key␣
↪variables

# sample weight (WGTHH, individual weight)
selectedWeightVarIND = c('WGTHH')

# Household ID
selectedHouseholdID = c('IDH')

# Variables not suitable for release in PUF (IND level)
varsNotToBeReleasedIND <- c("ATSCHOOL", "EDYRSCURRAT")

# All individual level variables
INDVars <- c(selectedKeyVarsIND)

# Recombining anonymized HH data sets and individual level variables
indVars <- c("IDH", "IDP", selectedKeyVarsIND, "WGTHH") # HID and all non HH vars
```

(continues on next page)

(continued from previous page)

```
fileInd <- file[indVars] # subset of file without HHVars
fileCombined <- merge(HHmanip, fileInd, by.x = c('IDH'))
fileCombined <- fileCombined[order(fileCombined[,'IDH'],  fileCombined[,'IDP']),]

dim(fileCombined)
## [1] 10068    44

# SDC objects with only IND level variables
sdcCombined <- createSdcObj(dat = fileCombined, keyVars = c(selectedKeyVarsIND),
                            weightVar = selectedWeightVarIND, hhId =␣
↪selectedHouseholdID)

# SDC objects with both HH and IND level variables
sdcCombinedAll <- createSdcObj(dat = fileCombined,
                               keyVars = c(selectedKeyVarsIND, selectedKeyVarsHH ),
                               weightVar = selectedWeightVarIND, hhId =␣
↪selectedHouseholdID)
```

12.2.11 Step 6b: Assessing disclosure risk (individual level)

As first measure, we evaluate the number of records violating k-anonymity at the levels 2, 3 and 5. Table 12.23 shows the number of violating individuals as well as the percentage of the total number of households. The second and third column refer to "sdcCombined" and the fourth and fifth column to "sdcCombinedAll". We see that combining the individual level and household level variables leads to a large increase in the number of k-anonymity violators. The choice not to include the household level variables is pragmatically driven by the computation time and can be justified by the different type of variables on the household level and individual level. One could assume that these variables are not available in the same dataset and can therefore not simultaneously be used by an intruder to re-identify individuals.

Table 12.23: Number of records violating k-anonimity

.	sdcCombined		sdcCombinedAll	
k-anonymity	Number of records violating	Percentage of total records	Number of records violating	Percentage of total records
2	0	0.0 %	4,048	40.2 %
3	167	1.7 %	6,107	60.7 %
5	463	4.6 %	8,292	82.4 %

The global hierarchical risk measure is 0.095%, which corresponds to approximately 10 expected re-identifications. We use here the hierarchical risk measure, since the re-identification of a single household member would lead to the re-identification of the other members of the same household too. This number is low compared to the number of k-anonymity violations, due to the high sample weights, which protect the data already to a large extent. Only 24 observations have an individual hierarchical risk higher than 1%, with a maximum of 1.17%. This is mainly because of the lower sample weights of these records. Listing 12.42 shows how to retrieve these measures in *R*.

Listing 12.42: Risk measures before anonymization

```
numIND <- length(fileCombined[,1]) # number of households

# Number of observations violating k-anonymity
print(sdcCombined)
## Infos on 2/3-Anonymity:
##
## Number of observations violating
```

(continues on next page)

(continued from previous page)

```
##  - 2-anonymity: 0
##  - 3-anonymity: 167
##
## Percentage of observations violating
##  - 2-anonymity: 0.000 %
##  - 3-anonymity: 1.659 %
## ------------------------------------------------------------------------

# Calculate sample frequencies and count number of obs. violating k(3,5) - anonymity
kAnon5 <- sum(sdcCombined@risk$individual[,2] 5)
kAnon5
## [1] 463

# As percentage of total
kAnon5 / numIND
## [1] 0.04598729

# Global risk on individual level
print(sdcCombined, 'risk')
## Risk measures:
##
## Number of observations with higher risk than the main part of the data: 0
## Expected number of re-identifications: 1.69 (0.02 %)
##
## Information on hierarchical risk:
## Expected number of re-identifications: 9.57 (0.10 %)
## ------------------------------------------------------------------------

# Number of observation with relatively high risk
dim(fileCombined[sdcCombined@risk$individual[, "hier_risk"] > 0.01,])
## [1] 24 44

# Highest individual risk
max(sdcCombined@risk$individual[, "hier_risk"])
## [1] 0.01169091
```

12.2.12 Step 7b: Assessing utility measures (individual level)

We evaluate the utility measures as discussed in Step 5 based on the raw data (before applying any anonymization measures). The results are presented in Step 10b together with the values after anonymization to allow for direct comparison.

12.2.13 Step 8b: Choice and application of SDC methods (individual level)

In this step, we discuss four different techniques used for anonymization: 1) removing variables from the dataset to be released, 2) recoding of categorical variables to reduce the level of detail, 3) local suppression to achieve the required level of k-anonymity, 4) randomization of the order of the records in the file. Finally, we discuss some alternative options for treating the household structure in the dataset.

Removing variables

Additional to the variables removed from the dataset for the SUF release (see case study 1), we further reduce the number of variables in the dataset to be released. This is normal practice for PUF releases. Sensitive or identifying

variables are removed, which allows to release other variables at a more detailed level. In a PUF release, the set of key variables should be limited.

In our case, we decide to remove at the individual level the variables "EDYRSCURRAT", as this variable is too identifying (identifies whether there are school-going children in the household). We keep the variable "EDUCY" (highest level of education attended) for information on education.

Note: As an alternative to removing the variables from the dataset, one could also set all values to missing. This would allow the user to see the structure and variables contained in the SUF file.

Recoding

As noted before, PUF users require a lower level of information and therefore we can recode the key variables even further to reduce the disclosure risk. The recoding of variables in case study 1 is not sufficient for a PUF release. Therefore, we recode most of the categorical key variables from Table 12.19 to reduce the risk and number of necessary suppressions by local suppression. Table 12.24 gives an overview of the recodes made. All new categories are formed with the needs of the data user in mind. Listing 12.43 shows how to do this in *R* and also shows value labels and the univariate tabulations of these variables before and after recoding.

Table 12.24: Overview of recodes of categorical variables at individual level

Variable	Recoding
REL (relation to household head)	recode 'Father/Mother', ' Grandchild', 'Son/Daughter in law', 'Other relative' to 'Other relative' and recode 'Domestic help' and 'Non-relative' to 'Other'
MARITAL (marital status)	recode 'Married monogamous', 'Married polygamous', 'Common law, union coutumiere, union libre, living together' to 'Married/living together' and 'Divorced/Separated' and 'Widowed' to 'Divorced/Separated/Widowed'
AGEYRS (age in completed years)	recode values under 15 to 7 (other values have been recoded for SUF)
EDUCY (highest level of education completed)	recode 'Completed lower secondary (or post-primary vocational education) but less than completed upper secondary', 'Completed upper secondary (or extended vocational/technical education)', 'Post secondary technical' and 'University and higher' to 'Completed lower secondary or higher'
INDUSTRY1	recode to 'primary', 'secondary' and 'tertiary'

Listing 12.43: Recoding the categorical and continuous variables

```
# Recode REL (relation to household head)
table(sdcCombined@manipKeyVars$REL, useNA = "ifany")
##
##    1    2    3    4    5    6    7    8    9 <NA>
## 1698 1319 4933   52  765   54  817   40   63  327

# 1 - Head, 2 - Spouse, 3 - Child, 4 - Father/Mother, 5 - Grandchild, 6 - Son/
↪Daughter in law
# 7 - Other relative, 8 - Domestic help, 9 - Non-relative
sdcCombined <- groupVars(sdcCombined, var = "REL", before = c("4", "5", "6", "7"),
                         after = c("7", "7", "7", "7")) # other relative
sdcCombined <- groupVars(sdcCombined, var = "REL", before = c("8", "9"),
                         after = c("9", "9")) # other

```

(continues on next page)

(continued from previous page)

```
table(sdcCombined@manipKeyVars$REL, useNA = "ifany")
##
##    1    2    3    7    9 <NA>
## 1698 1319 4933 1688  103  327

# Recode MARITAL (marital status)
table(sdcCombined@manipKeyVars$MARITAL, useNA = "ifany")

##
##    1    2    3    4    5    6 <NA>
## 3542 2141  415  295  330  329 3016

# 1 - Never married, 2 - Married monogamous, 3 - Married polygamous,
# 4 - Common law, union coutumiere, union libre, living together, 5 - Divorced/
↪Separated,
# 6 - Widowed
sdcCombined <- groupVars(sdcCombined, var = "MARITAL", before = c("2", "3", "4"),
                         after = c("2", "2", "2")) # married/living together
sdcCombined <- groupVars(sdcCombined, var = "MARITAL", before = c("5", "6"),
                         after = c("9", "9")) # divorced/seperated/widowed*

table(sdcCombined@manipKeyVars$MARITAL, useNA = "ifany")
##
##    1    2    9 <NA>
## 3542 2851  659 3016

# Recode AGEYRS (0-15 years)
table(sdcCombined@manipKeyVars$AGEYRS, useNA = "ifany")
##
##    0    1    2    3    4    5    6    7    8    9   10   11   12   13   14
##  311  367  340  332  260  334  344  297  344  281  336  297  326  299  263
##   20   30   40   50   60   65 <NA>
## 1847 1220  889  554  314  325  188

sdcCombined <- groupVars(sdcCombined, var = "AGEYRS",
                         before = c("0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
                                    "10", "11", "12", "13", "14"),
                         after = rep("7", 15))

table(sdcCombined@manipKeyVars$AGEYRS, useNA = "ifany")
##
##    7   20   30   40   50   60   65 <NA>
## 4731 1847 1220  889  554  314  325  188

sdcCombined <- calcRisks(sdcCombined)
# Recode EDUCY (highest level of educ compl)
table(sdcCombined@manipKeyVars$EDUCY, useNA = "ifany")
##
##    0    1    2    3    4    5    6 <NA>
## 1582 4755 1062  330  139   46  104 2050

# 0 - No education, 1 - Pre-school/ Primary not completed,
# 2 -  Completed primary, but less than completed lower secondary
# 3 - Completed lower secondary (or post-primary vocational education)
#     but less than completed upper secondary
# 4 - Completed upper secondary (or extended vocational/technical education),
# 5 - Post secondary technical
```

(continues on next page)

(continued from previous page)

```
# 6 - University and higher
sdcCombined <- groupVars(sdcCombined, var = "EDUCY", before = c("3", "4", "5", "6"),
                         after = c("3", "3", "3", "3")) # completed lower secondary or higher
table(sdcCombined@manipKeyVars$EDUCY, useNA = "ifany")
##
##    0    1    2    3 <NA>
## 1582 4755 1062  619 2050

# Recode INDUSTRY1 ()
table(sdcCombined@manipKeyVars$INDUSTRY1, useNA = "ifany")
##
##    1    2    3    4    5    6    7    8    9   10 <NA>
## 5300   16  153    2   93  484   95   17   70  292 3546

# 1 - Agriculture and Fishing, 2 - Mining, 3 - Manufacturing, 4 -  Electricity and Utilities
# 5 - Construction, 6 - Commerce, 7 - Transportation, Storage and  Communication, 8 - Financial, Insurance and Real Estate
# 9 - Services: Public Administration, 10 - Other Services, 11 - Unspecified
sdcCombined <- groupVars(sdcCombined, var = "INDUSTRY1", before = c("1", "2"),
                         after = c("1", "1")) # primary
sdcCombined <- groupVars(sdcCombined, var = "INDUSTRY1", before = c("3", "4", "5"),
                         after = c("2", "2", "2")) # secondary
sdcCombined <- groupVars(sdcCombined, var = "INDUSTRY1", before = c("6", "7", "8", "9", "10"),
                         after = c("3", "3", "3", "3", "3")) # tertiary
table(sdcCombined@manipKeyVars$INDUSTRY1, useNA = "ifany")
##
##    1    2    3 <NA>
## 5316  248  958 3546
```

Local suppression

The recoding has reduced the risk already considerably. We use local suppression to achieve the required level of k-anonymity. Generally, the required level of k-anonymity for PUF files is 3 or 5. In this case study, we require 5-anonimity. Listing 12.44 shows the suppression pattern without specifying an importance vector. All suppressions are made in the variable "AGEYRS". This is the variable with the highest number of different values, and hence considered first by the algorithm. We try different suppression patterns by specifying importance vectors, but we decide that the pattern without importance vector yields the best result. This is also the result with the lowest total number of suppressions. Less than 1 percent suppression in the age variable is acceptable. We could reduce this number by further recoding the variable "AGEYRS".

Listing 12.44: Local suppression to reach 5-anonimity

```
# Local suppression without importance vector
sdcCombined <- localSuppression(sdcCombined, k = 5, importance = NULL)

# Number of suppressions per variable
print(sdcCombined, "ls")
## Local Suppression:
##      KeyVar | Suppressions (#) | Suppressions (%)
##      GENDER |                 0 |            0.000
##         REL |                 0 |            0.000
##    MARITAL |                 0 |            0.000
##      AGEYRS |                91 |            0.904
```

(continues on next page)

(continued from previous page)

```
##       EDUCY |                 0 |             0.000
##   INDUSTRY1 |                 0 |             0.000
```

Randomization of order of records

The records in the dataset are ordered by region and household ID. There is a certain geographical order of the households within the regions, due to the way the households IDs were assigned. Intruders could reconstruct suppressed values by using this structure. To prevent this, we randomly reorder the records within the regions. Listing 12.45 shows how to do this in *R*. We first count the number of records per region

Note: Some records have their region value suppressed, so we include the count of NAs.

Subsequently, we draw randomly household IDs, in such way that the regional division is respected. Finally, we sort the file by the new, randomized, individual ID ("IDP"). Households with suppressed values for "REGION" will be last in the reordered file. Before randomizing the order, we extract the data from the *sdcMicro* object "sdcCombined" as shown in Listing 12.45.

Listing 12.45: Randomizing the order of records within regions

```
# Randomize order of households dataAnon and recode IDH to random
number (sort file by region)
set.seed(97254)
# Sort by region
dataAnon <- dataAnon[order(dataAnon$REGION),]

# Number of households per region
hhperregion <- table(dataAnon[match(unique(dataAnon$IDH), dataAnon$IDH), "REGION"],
                     useNA = "ifany")

# Randomized IDH (household ID)
randomHHid <- c(sample(1:hhperregion[1], hhperregion[1]),
                unlist(lapply(1:(length(hhperregion)-1),
                              function(i){sample((sum(hhperregion[1:i]) + 1):␣
↪sum(hhperregion[1:(i+1)]), hhperregion[(i+1)])})))

dataAnon$IDH <- rep(randomHHid, table(dataAnon$IDH)[match(unique(dataAnon$IDH),
                                    as.numeric(names(table(dataAnon$IDH))))])

# Sort by IDH (and region)
dataAnon <- dataAnon[order(dataAnon$IDH),]
```

Alternative options for dealing with household structure

In Step 6b we compared the disclosure risk for two cases: one with only individual level key variables and another with individual level and household level key variables combined. We decided to use only the individual level key variables to reduce the computation time and justified this choice by arguing that intruders cannot use household and individual level variables simultaneously. This might not always be the case. Therefore we explore other options to reduce the risk when taking both individual level and household level variables into account. We present two options: removing the household structure; and using options in the local suppression algorithm.

Removing household structure

We consider the risk emanating from the household structure in the dataset to be very high. We can remove the hierarchical household structure completely and treat all variables at the individual level. This entails, besides removing the household id ("IDH"), also treating variables that could be used for reconstructing households. These are, for instance, "REL" (relation to household head), "HHSIZE" (household size), and any of the household level variables, such as

income and expenditure. However, not all household level variables need to be treated. For example, "REGION" is a household level variable, but the probability that this variable leads to the reconstruction of a household is low. Also, we need to reorder the records in the file, as they are sorted by households. Note that by removing the household structure, we interpret all variables as individual level variables for measuring disclosure risk. This leads to a lower need for recoding and suppression, since the hierarchical risk disappears. The reason why we did not opt for this approach is the loss of utility for the user. The household structure is an important feature of the data, and should be kept in the PUF file.

Using different options for local suppression

The long running time is mainly due to the local suppression algorithm. In the Section Local suppression we discuss options to reduce the running time of the local suppression in case of many key variables. The all-m approach reduces the running time by first considering subsets of the complete set of key variables. This reduces the complexity of the problem and leads to lower computation times. However, the total number of suppressions made is likely to be higher. Also, if not explicitly specified, it is not guaranteed that the required level for k-anonymity is automatically achieved on the complete set of key variables. It is therefore important to check the results.

12.2.14 Step 9b: Re-measure risk

We re-evaluate the risk measures selected in Step 6. Table 12.25 shows that local suppression, not surprisingly, has reduced the number of individuals violating 5-anonymity to 0. The global hierarchical risk was reduced to 0.02%, which corresponds to approximately 2 correct re-identifications. The highest individual hierarchical re-identification risk is 0.2%. These risk levels are acceptable for a PUF release. Furthermore, the recoding has removed any unusual combinations in the data.

Note: The risk may be underestimated by excluding the household level variables.

Table 12.25: k-anonymity violations

k-anonymity	Number of records violating	Percentage
2	0	0.0 %
3	0	0.0 %
5	0	0.0 %

12.2.15 Step 10b: Re-measure utility

We compare (cross-)tabulations before and after anonymization, which are illustrated in the *R* code to this case study. We note that due to the recoding in Step 8b, the detail in the variables is reduced. This reduces the number of necessary suppressions and is acceptable for a public use file.

12.2.16 Step 11: Audit and reporting

In the audit step, we check whether the data allow for reproduction of published figures from the original dataset and relationships between variables and other data characteristics are preserved in the anonymization process. In short, we check whether the dataset is valid for analytical purposes. There are no figures available that were published from the dataset and need to be reproducible from the anonymized data.

In Step 2, we explored the data characteristics and relationships between variables. These data characteristics and relationships have been mainly preserved, since we took them into account when choosing the appropriate anonymization methods. Since values of the variable "AGEYRS" were not perturbed, but only recoded and suppressed, we did not introduce unlikely combinations, such as a 60-year-old individual enrolled in primary education. Also, by separating

the anonymization process into two parts, one for household-level variables and one for individual-level variables, the values of variables measured at the household level agree for all members of each household.

Furthermore, we drafted two reports, internal and external, on the anonymization of the case study dataset. The internal report includes the methods used, the risk before and after anonymization as well as the reasons for the selected methods and their parameters. The external report focuses on the changes in the data and the loss in utility. Focus here should be on the number of suppressions as well as the perturbative methods (PRAM). This is described in the previous steps.

Note: When creating a PUF, it is inevitable that there will be a loss of information and it is very important for the users to be aware of these changes and release them in a report that accompanies the data.

Appendix C provides examples of an internal and external report of the anonymization process of this dataset. Depending on the users and readers of the reports, the content may differ.

Note: The report() function in sdcMicro is at this point not useful, since this will only report on the SDC measures in the second case study.

However, the report should contain the entire process, including the measures applied in case study 1.

12.2.17 Step 12: Data release

The final step is the release of the anonymized dataset together with the external report. Listing 12.46 shows how to export the anonymized dataset as *STATA* file. The Section Read functions in R presents functions for exporting files in other data formats.

Listing 12.46: Exporting the anonymized PUF file

```
# Create STATA file
write.dta(dataframe = dataAnon, file= 'Case2DataAnon.dta', convert.dates=TRUE)
```

BIBLIOGRAPHY

[DuBo10] Dupriez, O., & Boyko, E. (2010). **Dissemination of Microdata Files; Principles, Procedures and Practices.** International Household Survey Network (IHSN).

[HDFG12] Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Nordholt, E. S., Spicer, K., et al. (2012). **Statistical Disclosure Control.** Chichester, UK: John Wiley & Sons Ltd.

[DuBo10] Dupriez, O., & Boyko, E. (2010). **Dissemination of Microdata Files; Principles, Procedures and Practices.** International Household Survey Network (IHSN).

[DoTo03] Domingo-Ferrer, J., & Torra, V. (2003). **Disclosure Risk Assesment in Statistical Microdata Protection via Advanced Record Linkage.** Statistics and Computing 13 (4), 343-354

[ElMa03] Elliot , M. J., & Manning, A. M. (2003). **Using DIS to Modify the Classification of Special Uniques.** Invited Paper. Joint ECE/Eurostat Work Session on Statistical Data Confidentiality. Luxemboug 2-9 April 2003.

[ElMF02] Elliot, M. J., Manning, A. M., & Ford, R. W. (2002). **A Computational Algorithm for Handling the Special Uniques Problem.** International Journal of Uncertainty, Fuzziness and Knowledge Based System , 10 (5), 493-509.

[ELMP10] Elliot, M. J., Lomax, S., Mackey, E. & Purdam, K. (2010) **Data Environment Analysismand the Key Variable Mapping System.** In Privacy in Databases, PSD 2010 (ed. Domingo-Ferrer, J. & Magkos, E.), vol. 6344 of Lecture Notes in Computer Science, pp. 138-145. Berlin/Heidelberg: Springer

[ElSD98] Elliot, M.J., Skinner, C.J. & Dale, A. (1998) **Special Uniques, Random Uniques, and Sticky Populations: Some Counterintuitive Effects of Geographical Detail on Disclosure Risk** Reasearch in Official Statistics 1(2), pp. 53-67

[EMMG05] Elliot, M. J., Manning, A., Mayes, K., Gurd, J., & Bane, M. (2005). **SUDA: A Program for Detecting Special Uniques.** Joint UNECE/Eurostat Work Session on Statistical Data Confidentiality. Geneva.

[HDFG12] Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Nordholt, E. S., Spicer, K., et al. (2012). **Statistical Disclosure Control.** Chichester, UK: John Wiley & Sons Ltd.

[Lamb93] Lambert, D. (1993). **Measures of Disclosure Risk and Harm.** Journal of Official Statistics , 9 (2), 313-331.

[MaHK08] Manning, A. M., Haglin, D. J., & Keane, J. A. (2008). **A Recursive Search Algorithm for Statistical Disclosure Assessment.** Data Mining and Knowledge Discovery , 16 (2), 165-196.

[MKGV07] Machanavajjhala, A., Kifer, D., Gehrke, J., & Venkitasubramaniam, M. (2007). **L-diversity: Privacy Beyond K-anonymity.** ACM Trans. Knowl. Discov. Data , 1 (Article 3) (1556-4681).

[TeMe08] Templ, M. & Meindl, B. (2008) **Robust Statistics Meets SDC: New Disclosure Risk Measures for Continuous Microdata Masking.** In Privacy in Statistical Databases, PSD 2008 (eds. Domingo-Ferrer J. and Saygin Y.), vol. 5262 of Lecture Notes in Computer Science, pp. 177-189. Berlin/Heidelberg: Springer.

[TeMK14] Templ, M., Meindl, B., & Kowarik, A. (2014, August). **Tutorial for SDCMicroGUI.** Retrieved from International Household Survey Network (IHSN): http://www.ihsn.org/home/software/disclosure-control-toolbox

[TMKC14] Templ, M., Meindl, B., Kowarik, A., & Chen, S. (2014, August 1). **Introduction to Statistical Disclosure Control (SDC).** Retrieved November 13, 2014, from http://www.ihsn.org/home/software/disclosure-control-toolbox.

[BCRZ13] Burgert, C. R., Colston, J., Roy, T., & Zachary, B. (2013). **Geographic Displacement Procedure and Georeferenced Data Release Policy for the Demographic and Health Surveys.** DHS Spatial Analysis Report No. 7.

[Bran02] Brand, R. (2002). **Microdata Protection through Noise Addition.** In J. Domingo-Ferrer (Ed.), Inference Control in Statistical Databases - From Theory to Practice (Vol. Lecture Notes in Computer Science Series Volume 2316, pp. 97-116). Berlin Heidelberg, Germany: Springer.

[DMOT02] Domingo-Ferrer, J., Mateo-Sanz, J.M., Oganian, A. & Torres, A. **On the Security of Microaggregation with Individual Ranking: Analytics Attacks.** International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(5), pp. 477-492.

[DoTo01a] Domingo-Ferrer, J., & Torra, V. (2001). **A Quantitative Comparison of Disclosure Control Methods for Microdata.** In P. Doyle, J. Lane, J. Theeuwes, & L. Zayatz (Eds.), Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies (pp. 111-133). Amsterdam, North-Holland: Elsevier Science.

[DoTo05] Domingo-Ferrer, J., & Torra, V. (2005). **Ordinal, Continuous and Heterogeneous :math:'k'-anonimity through Microaggregation** Data Mining and Knowledge Discovery 11(2), pp. 195-212.

[Drec11] Drechsler, J. (2011). **Synthetic Datasets for Statistical Disclosure Control.** Heidelberg/Berlin: Springer.

[HDFG12] Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Nordholt, E. S., Spicer, K., et al. (2012). **Statistical Disclosure Control.** Chichester, UK: John Wiley & Sons Ltd.

[HuDr15] Hu, J., & Drechsler, J. (2015). **Generating synthetic geocoding infromation for public release.** NTTS - Conferences on New Techniques and Technologies for Statistics. Brussels.

[KiWi03] Kim, J. J., & Winkler, W. W. (2003, April 17). **Multiplicative Noise for Masking Continuous Data.** Research Report Series.

[MuSa06] Muralidhar, K., & Sarathy, R. (2006). **Data Shuffling- A New Masking Approach for Numerical Data.** Management Science , 658-670.

[TeMK14] Templ, M., Meindl, B., & Kowarik, A. (2014, August). **Tutorial for SDCMicroGUI.** Retrieved from International Household Survey Network (IHSN): http://www.ihsn.org/home/software/disclosure-control-toolbox

[TMKC14] Templ, M., Meindl, B., Kowarik, A., & Chen, S. (2014, August 1). **Introduction to Statistical Disclosure Control (SDC).** Retrieved July 9, 2018, from http://www.ihsn.org/home/software/disclosure-control-toolbox.

[Wolf15] de Wolf, P.-P. (2015). **Public Use Files of EU-SILC and EU-LFS data.**

[DoTo01b] Domingo-Ferrer, J., & Torra, V. (2001). **Disclosure Protection Methods and Information Loss for Microdata.** In P. Doyle, J. Lane, J. Theeuwes, & Z. L., Theory and Practical Applications for Statistical Agencies (pp. 91-110). Amsterdam.

[HDFG12] Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Nordholt, E. S., Spicer, K., et al. (2012). **Statistical Disclosure Control.** Chichester, UK: John Wiley & Sons Ltd.

[TMKC14] Templ, M., Meindl, B., Kowarik, A., & Chen, S. (2014, August 1). **Introduction to Statistical Disclosure Control (SDC).** Retrieved July 9, 2018, from http://www.ihsn.org/home/software/disclosure-control-toolbox.

[YaWC02] Yancey, W. W., Winkler, W. E., & Creecy, R. H. (2002). **Disclosure Risk Assessment in Perturbative Microdata Protection.** Research Report Series , Statistics 2002-01.

[HDFG12] Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Nordholt, E. S., Spicer, K., et al. (2012). **Statistical Disclosure Control.** Chichester, UK: John Wiley & Sons Ltd.

[DuBo10] Dupriez, O., & Boyko, E. (2010). **Dissemination of Microdata Files; Principles, Procedures and Practices.** International Household Survey Network (IHSN).