Advanced Survey Statistics: Disclosure Control

Part 8: Walkthrough / Workflow

Matthias Templ

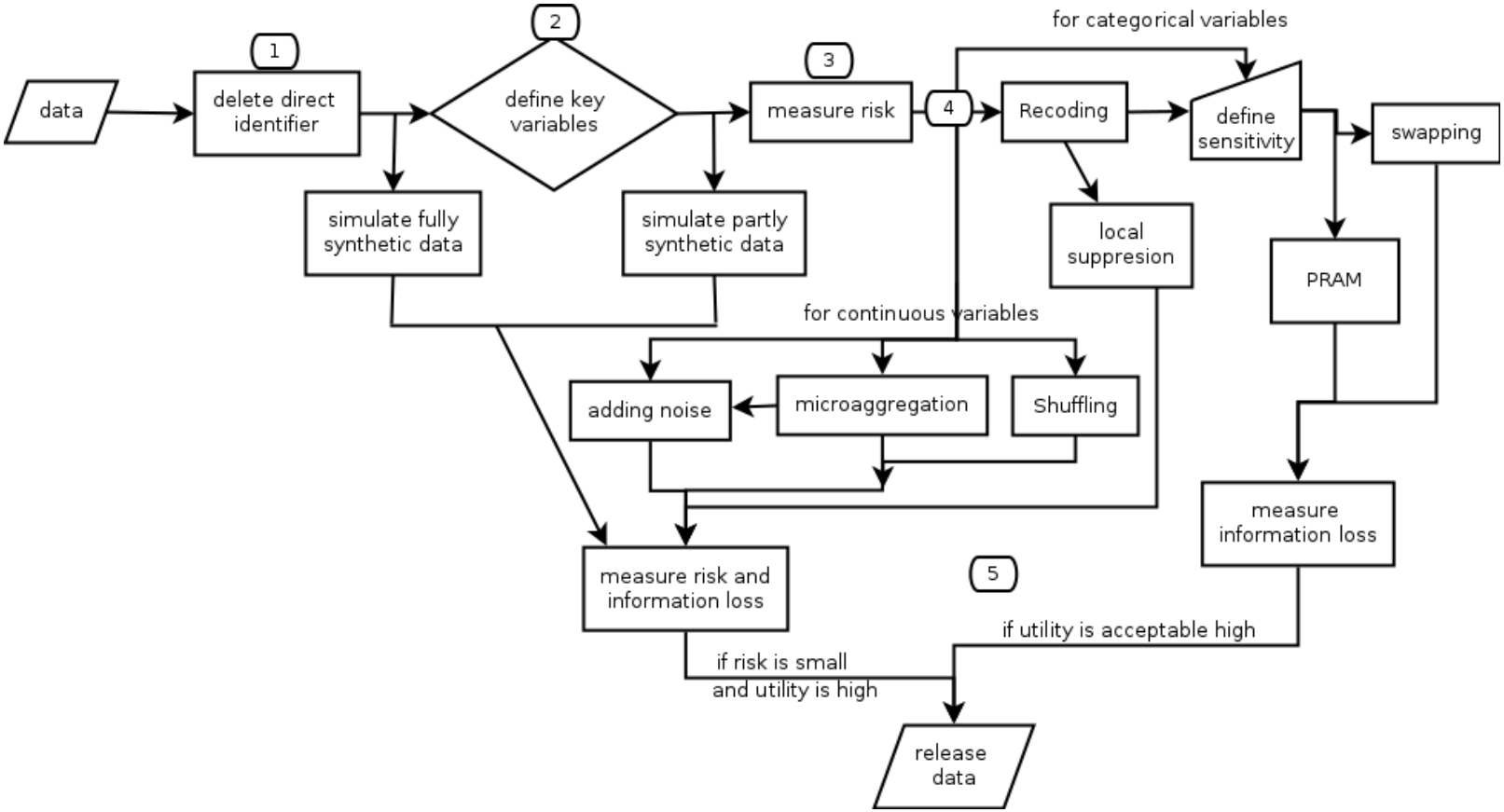
Institut für Datenanalyse und Prozessdesign
School of Engineering
Zürcher Hochschule für Angewandte Wissenschaften

FU-Berlin, 2019

Zürcher Hochschule
für Angewandte Wissenschaften

School of Engineering
IDP Institut für Datenanalyse
und Prozessdesign

Workflow, simplified



First step

Pseudo-anonymisation or **deletion** of key variables.

Excursus: Pseudo-anonmyisation

Excursus Pseudo-anonymization

Hash function

- A cryptological **hash function** or cryptographic hash function is a special form of a hash function.
- It is virtually impossible to find two different input values that give an identical hash value. This is known as collision security.
- The original content can not be generated by users
- The **hash value** (with predefined length) itself represents the result which was calculated by means of a hash function, in most cases the hash value is encoded as a hexadecimal character string

Excursus Pseudo-anonymization

A hash value of the name „Matthias Templ“ might look like this:

```
# devtools::install_github("paulhendricks/anonymizer")
library("anonymizer")
name <- "Matthias Templ"
myhash <- hash(name, .algo = "sha256", .seed = 123)
myhash
```

```
## [1] "6f25de7af3ef4795a0d19f34eee7baa422f71463ab9bd244ab6
```

- Hash-algorithm, default and recommended: SHA-256 algorithm from NIST.
- Others can be chosen optionally: sha1, crc32, sha512, xxhash32, xxhash64 and murmur32.

Excursus Pseudo-anonymization

Salt

- Salt, in cryptography, refers to a random string of characters appended to a given plaintext **prior** to use as a hash function input.
- The salt used in pseudo-anonymization is stored in a database along with the resulting hash value.
- The use of salt increases the effort of these attacks significantly, especially dictionary attacks or general attacks with existing registers of names, social security numbers, etc. are made more difficult, since no longer in a list of hash values (hash table) the corresponding plain text (password) are looked up can, but for every plaintext hash (plaintext + salt) = known hash must be checked.

Excursus Pseudo-anonymization

Salt in R

```
name_salt <- salt("Matthias Templ",
                  .n_char = 5,
                  .seed = 123)
name_salt
```

```
## [1] "osncjMatthias Templosncj"
```

```
myhash_salt <- hash(name_salt, .seed = 123)
myhash_salt
```

```
## [1] "35aa847ac69ce773bccdcff2e20ec934aae136002145de9a665
```

Excursus Pseudo-anonymization

Unsalt

Given the starting value of the random number generator used, an unsalt can be made at any time when the seed is known.

```
unsalt(salt(name_salt, .seed = 123), .seed = 123)
```

```
## [1] "Matthias Templ"
```

Excursus Pseudo-anonymization

User- and institution-specific hashing

- Each user receives different hashes for the same ID (user-specific hashing)
- If one user requests some additional data, he also gets different hashes (project-specific hashing)
- Different institutions receives different hashs for the same ID's
- Only one central location (secure server) of the data provider can switch from the hash to the actual ID and **only this location can link data via the ID**.
- Data recipients can never link data sets.

Excursus Pseudo-anonymization, Toy example

```
x1 <- data.frame("names" = c("Matthias Templ",
                             "Beat Wolf"),
                 "costs" = c(37845, 36231))
x1
```

```
##             names costs
## 1 Matthias Templ 37845
## 2      Beat Wolf 36231
```

```
x2 <- data.frame("names" = c("Eva Collins",
                             "Beat Wolf"),
                 "religion" = c("kath", "prot"))
x2
```

```
##          names religion
## 1 Eva Collins     kath
## 2   Beat Wolf     prot
```

Excursus Pseudo-anonymization, Toy example

```
m <- merge(x1, x2, by = "names", all = TRUE)
m # Merge is succesful
```

```
##             names costs religion
## 1      Beat Wolf 36231     prot
## 2 Matthias Templ 37845     <NA>
## 3    Eva Collins    NA     kath
```

Same ID's do not help

```
x1$names <- c("3000823", "3423525")
x2$names <- c("4634566","3423525")
merge(x1, x2, by = "names", all = TRUE)
```

```
##     names costs religion
## 1 3000823 37845     <NA>
## 2 3423525 36231     prot
## 3 4634566    NA     kath
```

Excursus Pseudo-anonymization, Toy example

```
m$salt_names1 <- salt(m$names, .seed = 123)
m$salt_names2 <- salt(m$names, .seed = 1234)
m$hash_salt1 <- hash(m$salt_names1, .seed = 123,
                     .algo = "xxhash32")
m$hash_salt2 <- hash(m$salt_names2, .seed = 123,
                     .algo = "xxhash32")

m <- m[, c(1,4,5,6,7,2,3)]
m
```

```
##             names                salt_names1                 s
## 1      Beat Wolf      osncjBeat Wolfosncj      pzvelBeat
## 2 Matthias Templ osncjMatthias Templosncj pzvelMatthias
## 3    Eva Collins    osncjEva Collinsosncj    pzvelEva Co
##   hash_salt1 hash_salt2 costs religion
## 1   203fd876   4eca3ad3 36231     prot
## 2   a320bf4f   0631ddee 37845     <NA>
```

Excursus Pseudo-anonymization, Toy example

```
x1 <- m[1:2, c(4,6)] # first data set
x1
```

```
##   hash_salt1 costs
## 1   203fd876 36231
## 2   a320bf4f 37845
```

```
x2 <- m[2:3, c(5,7)] # second data set
x2
```

```
##   hash_salt2 religion
## 2   0631ddee     <NA>
## 3   b966fc19     kath
```

Merge isn't possible anymore (can be done only by the data holder who applied salt + hash)

Workflow

- For non-synthetic methods, the choice of key variables is very important. This means asking „what is the knowledge of an attacker“? In other words, what possible databases with intersectional populations may be available to someone who receives the data.
- For key categorical variables, prior to using SDC techniques, the re-identification risks of the data should be modeled and estimated. This allows the identification of records with high re-identification risks, e.g. those that violate k-anonymity (typically 3-anonymity) or assess high-risk observations using the individual risk approach.
- For continuous key variables, there are other methods for determining re-identification risk. Best is to think on possible matching / record linkage.
- Also, a global re-identification risk should be estimated using either a log-linear modeling approach or the sum of individual risks.

Workflow - applying anonymization methods

- For categorical key variables apply global recoding and local suppression to establish k anonymity, or alternatively, swapping techniques could be used.
- Continuous key variables could be treated by shuffling, microaggregation, or the addition of noise, or a combination of these techniques
- Each time an SDC technique is used, report the risk of re-identification and the increase of information loss
- The anonymization process is typically **iterative** after the scenario (key variable) is set. Different anonymization methods are tried until the risk of re-identification has been decisively reduced and at the same time the loss of information is considerable low.
- For both categorical and continuous key variables, the loss of information should not be valued only by *general purpose measures*, but should also be quantified by estimating indicators or models

How to determine the key variables

The basis for estimating the risk is the disclosure scenario.

- determining key variables is challenging because there are no clear rules and each variable potentially belongs to key variables.
- recommended: consider multiple disclosure scenarios and discuss them with specialists to find out which scenario is most likely and most realistic.
- A common scenario is that a data user associates the data provided to him or her with external data sources, so an important step is to inventory what other data sources are available and linkable.
  - For example, age, gender, and place of residence will probably be in publicly accessible or at least commercially available databases.
- In addition, sensitive variables containing confidential information should also be identified in advance.
  - for example, variables about the state of the insured person.

Re-identification risk: when is it small enough?

The acceptable level of risk depends on many factors.

- public-use files should have much less disclosure risk than scientific-use files, which is restricted to certain users under certain conditions.
- a dataset containing sensitive information such as HIV / AIDS information may require greater intervention for anonymisation, compared to data with less sensitive information.
- It is recommended to discuss issues regarding the level of risk along with management, lawyers and experts.
- The risk is never 0, but this is not required (de-facto anonym.).
- Look carefully at the distribution of individual risks.
  - if individual observations with much higher risk than the rest of the observations (then do something with these highly risky observations)
  - an individual risk of eg 0.1 would be too high with 10% prob. the observation would be re-identified (given the chosen scenario), whereas 0.005 is very low.

Re-identification risk: when is it small enough?

- When values are swapped the tradtional approach to estimate the risk is not possible and thus, the question of the size of the swap rate is difficult.
- What is the probability that a match will contain the right data? Is it enough to know that in a successful match at swapping rate, 10% of data users do not have true values.
- Or should the swapping rate be larger? This is also a socio-political, ethical and scientific-ethical decision that can not be made on the basis of statistical considerations alone
  - Swap rates of 7% were used in the Austrian Census (without reference),
  - 2% and 5% in the UK Census
  - and different swapping rates for different data recipients from 0 to more than 26% in the UK Census 2011.

Differences for data delivery externally and internally?

Data protection must also be carried out when delivering data internally.

- The individual risk (calculated) that somebody can be identified is basically the same
- But it is clear that, in general, the risk for internal data delivery is smaller than externally, since (and only if)
  - the planned processing tasks are described internally, * the purposes of the processing may be better known,
  - risks to data storage can be minimized (the data may be less likely) can be stolen),
  - and it can also be logged who works when with the data, i.e. to be able to control this better than with external data delivery.

Which anonymization methods should be used

- The strength and weakness of each SDC method depends on the structure of the record and the key variables considered.
- The recommended approach is to apply different SDC methods with different parameter settings in an exploratory / iterative manner.

For categorical key variables,

- recoding is most commonly used.
- If the risks remain high after the recoding, local suppression can be used to further reduce the number of uniques

Which anonymization methods should be used

- the process of searching for good recoding is not based on a mathematical optimization function, but rather is an exploratory approach with the expertise about the data set to be anonymized. The new coding must be meaningful in terms of content and at the same time considerably reduce the risk.
  - For example, transcoding the age into four age groups ($ [0-19]; [20-39]; [40-59], [60-100] $) is not recommended if the exact age is important for an analysis. It is probably then better not to recode age but to add some noise to age.
- Another example is the change of regional information. If the researchers' goal is a spatial analysis, they can not work with data whose regional information has been heavily aggregated or swapped. On the other hand, if the data user's goal is to compare cantons, no more detailed information is needed at the community level.

Which anonymization methods should be used

If a data set contains a **large number of categorical key variables** and / or a large number of categories for the given key variables (eg, municipality ID) $\rightarrow$ recoding and suppression can lead to a high loss of information. Alternatives:

- apply PRAM
- generate synthetic data

For continuous variables,

- microaggregation is a recommended method.

Last steps:

- For high-risk individuals, further suppression and recoding should be undertaken.
- The sum of the individual risks as a global risk measure gives an impression of the global risk of a data set. This is also useful for comparing differently anonymized records.