MAT 4376 Foundations of Data Privacy

Rafał Kulik

Department of Mathematics and Statistics (University of Ottawa)

Lectures 5, 6

1. Anonymization methods
   - Classification of anonymization methods
   - Non-perturbative anonymization methods
   - Perturbative anonymization methods

During these lectures we will learn:

- basic anonymization methods;
- how these anonymization methods affect privacy.

We will not learn yet:

- how much information has been lost (data utility point of view);
- differential privacy.

Each data set may need a different anonymization method. It depends on the required level of privacy, type of disclosure, type of data (e.g. categorical/continuous).

Some resources

- A blog on SDC: https://sdcpractice.readthedocs.io/en/latest/measure_risk.html.
- MSc thesis of Chang Qu.
- PhD thesis of Devyani Biswal.

Classification of anonymization methods

First classification:

- **Non-perturbative methods** reduce the detail in the data by generalization or suppression of certain values (i.e., masking) without distorting the data structure. *Example:* k-anonymization.
- **Perturbative methods** perturb (i.e., alter) values to limit disclosure risk by creating uncertainty around the true values. *Example:* adding a random noise to each data (*does it always make sense?*).
    - Special case: adding noise to a query (**differential privacy**).

Classification of anonymization methods

Second classification:

- Probabilistic methods. *Example:* adding a random noise to each data (*does it always make sense?*).
- Deterministic methods. *Example:* k-anonymization.

Classification of anonymization methods

Method	Classification	Data Type
Recoding	non-perturb, deterministic	
Global recoding		cont. and categorical
Top and bottom coding		ordinal categorical continuous
Local suppression	non-perturb, determinitic	categorical
PRAM	perturbative, probabilistic	categorical
Micro aggregation	perturbative, probabilistic	continuous
Noise addition	perturbative, probabilistic	continuous
Shuffling	perturbative, probabilistic	continuous
Rank swapping	perturbative, probabilistic	continuous

Recoding (grouping)

Recoding is a **deterministic, non-perturbative method** used to decrease the number of distinct categories or values for a variable. Using the language we introduced before, we decrease the number of equivalence classes. 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.

Recoding - global recording

Global recoding is a **deterministic, non-perturbative method** that combines several categories (equivalence classes) of a categorical variable or constructs intervals for continuous variables. This reduces the number of equivalence classes available in the data and potentially the disclosure risk, especially for categories with few observations, but also it reduces the level of detail of information available to the analyst.

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.

Example: k-anonymization. Refer to Lectures 2-3.

Recoding - global recording

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

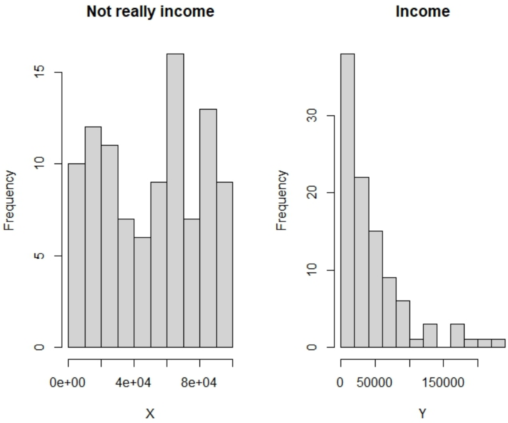
- The categories of **age** should be chosen so that they still allow data users to make calculations relevant for the subject being studied. For example, if indicators need to be calculated for children of school going ages 6 - 11 and 12 - 17, then it does not make any sense to group individuals into 0 - 10, 11 - 15, 16 - 18.
- **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.

Recoding - top and bottom

Top and bottom coding is a **deterministic, non-perturbative method** 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 (high quantiles)**).

We need to be able to detect outliers. Typically, we will calculate quantile(p) for p close to 1.

Recoding - top and bottom



Quantile function

Let X be a random variable and $F : \mathbb{R} \to [0,1]$ its cumulative distribution function (CDF): $F(x) = \mathbb{P}(X \le x)$. Then the quantile function $Q : (0,1) \to \mathbb{R}$ is

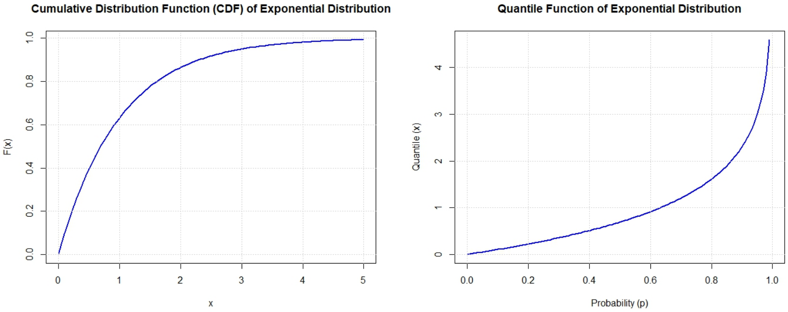
$$Q(u) = \inf\{x : F(x) \ge u\}.$$

Sometimes we can extend Q to $[0,1)$ or $(0,1]$ or $[0,1]$. If F is strictly increasing and continuous then Q is just the inverse function and $F(Q(u)) = u$, $u \in (0,1)$, or $Q(F(x)) = x$, $x \in \mathbb{R}$.

Example 1

Let $\lambda > 0$. If $F(x) = 1 - \exp(-\lambda x)$ for $x > 0$ and $F(x) = 0$ for $x \le 0$, then $Q(u) = -\frac{1}{\lambda}\log(1-u)$, $u \in [0,1)$.

CDF and Quantile function - Exponential



Quantile function

If F is not strictly increasing ...

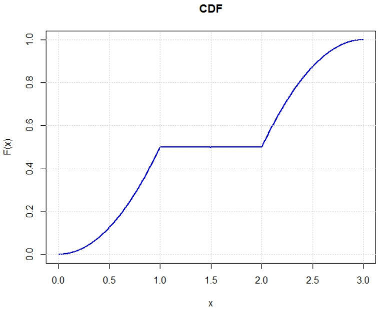
Example 2

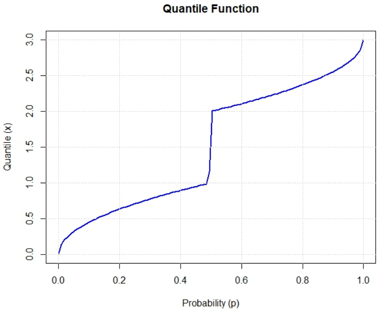
Assume that random variable X has the density

$$f(x) = 2x,\ x \in (0,1)\ ,\quad f(x) = 2(3-x),\ x \in (2,3)$$

and $f(x) = 0$ otherwise. Then F is constant between 2 and 3 and Q has a jump at 2.

CDF and Quantile function - special case





Quantiles and sample quantiles

Given data $X_1, \ldots, X_n$, the empirical cumulative distribution function (CDF):

$$\widehat{F}_n(x) = \frac{1}{n} \sum_{j=1}^{n} \mathbb{I}\{X_j \le x\} \,, \; x \in \mathbb{R} \,.$$

Note that $\widehat{F}_n$ is piecewise constant. Hence, the **sample quantile function**

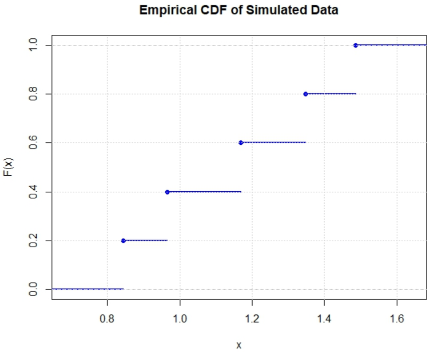
$$\widehat{Q}_n(u) = \inf\{x : \widehat{F}_n(x) \ge u\}$$

will have jumps at the ordered data points. In fact,

$$\widehat{Q}_n(i/n) = X_{(i)} \,, \; i = 1, \ldots, n$$

where $X_{(1)} \le \cdots \le X_{(n)}$ are the order statistics. Use `quantile(data,p)`, where $p \in (0,1)$ to obtain sample quantiles. Note that `R` use some special formula to calculate its own version $\widetilde{Q}_n$ of sample quantiles. For example, $\widetilde{Q}_n(0) = X_{(1)}$, $\widetilde{Q}_n(1) = X_{(n)}$ and there is some interpolation involved.

Quantiles and sample quantiles



Data: 0.8431497, 0.9658712, 1.1685290, 1.3480445, 1.4852758.
`quantile(observations,1)` returns 1.4852758,
`quantile(observations,0.75)` returns 1.3480445.

Recoding - example

Dataset of 10 individuals (age and gender as QIs). After recoding, I obtained 2-anonymization w.r.t age, but no 2-anonymization w.r.t. age+gender (Female, [40-49] is a singleton).

```
   age gender income
1   50   Male  39641 [50-59], Male
2   34   Male  89133 [30-39], Male
3   33   Male  82131 [30-39], Male
4   22 Female  44182 [20-29], Female
5   56   Male  45179 [50-59], Male
6   33   Male  57167 [30-39], Male
7   44   Male  39096 [40-49], Male
8   45   Male  60537 [40-49], Male
9   46 Female  86218 [40-49], Female ***
10  24 Female  37988 [20-29], Female
```

Recoding - example

I asked ChatGPT: *for the previous table, do recoding to obtain equivalence classes of size 2.* Note that the age of "46, Female" was generalized to [20-30]. It is *not* good. Also, ChatGPT used intervals (40,50], while I used [40,50). ChatGPT created a problem with another entry, "56, Male".

	age	gender	income	age_group	equivalence_class	
1	50	Male	39641	(40,50]	(40,50]_Male	
2	34	Male	89133	(30,40]	(30,40]_Male	
3	33	Male	82131	(30,40]	(30,40]_Male	
4	22	Female	44182	[20,30]	[20,30]_Female	
5	56	Male	45179	(40,50]	(40,50]_Male	*****
6	33	Male	57167	(30,40]	(30,40]_Male	
7	44	Male	39096	(40,50]	(40,50]_Male	
8	45	Male	60537	(40,50]	(40,50]_Male	
9	46	Female	86218	[20,30]	[20,30]_Female	****
10	24	Female	37988	[20,30]	[20,30]_Female	

Local suppression

It is common in surveys to encounter values for certain variables or combinations of quasi-identifiers that are shared by very few individuals. Sometimes recoding may not be feasible or gives undesirable answers. See the previous example. Often local suppression is used after reducing the number of keys in the data by recoding the appropriate variables. Suppression of values means that values of a variable are replaced by a missing value (NA in R).

Local suppression - example

	age_suppressed	gender_suppressed	income
1	50	Male	39641
2	34	Male	89133
3	33	Male	82131
4	22	Female	44182
5	NA	<NA>	45179
6	33	Male	57167
7	44	Male	39096
8	45	Male	60537
9	NA	<NA>	86218
10	24	Female	37988

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 Y is computed as

$$Y = AXB + C \ ,$$

where

- X is the original data $(n \times p)$-dimensional data set (n - the number of individuals, p - the number of variables);
- A is a $(m \times n)$-matrix used to transform the records;
- B is a $(p \times q)$-matrix to transform the variables;
- C is a $(m \times q)$-matrix with additive noise.

Perturbative methods

The type of perturbation depends on the type of data. In what follows, we will take our table and

- Add a random value from the set $\{-2, -1, 0, 1, 2\}$ to each age.
- Add a random value sampled uniformly from $[-5000, 5000]$ to each income.
- We swap randomly gender entries.

Perturbative methods - example

	age	gender	income	age_per	income_per	gender_per
1	50	Male	39641	50	42584.42	Male
2	34	Male	89133	32	88531.32	Female
3	33	Male	82131	32	84675.75	Male
4	22	Female	44182	24	45474.21	Male
5	56	Male	45179	58	47280.82	Male
6	33	Male	57167	34	52173.25	Female
7	44	Male	39096	46	38849.17	Male
8	45	Male	60537	44	57738.19	Male
9	46	Female	86218	44	85016.17	Male
10	24	Female	37988	22	39115.71	Female

Now, if I know that Mrs. Smith is "46, Female" (the one that caused problem before), I do not see her in the database.

Perturbative methods - word of caution

You will see sometimes

> *One advantage of perturbative methods is that the information loss is reduced, since no values will be suppressed, depending on the level of perturbation.*

This statement may be misleading. Especially, swapping records may completely destroy the dependence structure.

Perturbative methods - word of caution

Consider the example (we assume that age and income are the same in the original and the released dataset).

	age	gender	income	gender_swapped
1	50	Male	39641	Female
2	34	Male	39133	Female
3	33	Male	32131	Female
4	22	Female	74182	Male
5	56	Female	75179	Male
6	33	Female	87167	Male

Even though *marginally* all is perfect (the distribution of age, gender, income remain the same), even if all is perfect from the privacy perspective (if we know that Mr. Smith is 50, we will not get his income), we completely destroyed the relationship between gender and income.

PRAM

PRAM (Post RAndomization Method) is a perturbative method for categorical data. This method reclassifies the values of one or more variables, such that intruder that attempts 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.

Assume that the possible realizations of the random variables X_j lie in the set $\{a_i, i = 1, \ldots, M\}$, where a_i are real values. The basic idea is as follows: each of X_j's is transformed into Y_j according to the given transition probabilities:

$$p_{kl} = \mathbb{P}(Y_j = a_l \mid X_j = a_k) \; .$$

PRAM

The disclosure risk in PRAM is measured through *posterior odds*, that is, the relative probability that a rare score in the perturbed dataset Y corresponds with a rare score in the original dataset Y. These posterior odds should be small.

Example 3

We can illustrate this concept with a simple example. Suppose that the variable X_j represents gender of jth person with the possible values of 0 if you are male and 1 if you are female. PRAM can be applied to the gender variable so that $p_{kk} = 0.8$. Assume the database contains 1000 people, consisting of 500 men and 500 women. The expected perturbed database will also contain 500 men and women, but 100 men and 100 women would have had their gender swapped.

Comment: Extension to PRAM is called k-PRAM (combination of k-anonymity and PRAM) - see Section 3.3 in PhD of Devyani Biswal.

Microaggregation

Microaggregation is most suitable for continuous variables, but can be extended in some cases to categorical variables.

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 or median of that group. Microaggregation methods differ with respect to

- how the homogeneity of groups is defined,
- the algorithms used to find homogeneous groups,
- the determination of replacement values.

Microaggregation

In the univariate case, and also for ordinal categorical variables, formation of homogeneous groups is straightforward. Assume that we have data $X_1, \ldots, X_n$. Divide data into J groups $g_1, \ldots, g_J$ of indices of sizes $n_1, \ldots, n_J$. The group sizes can differ amongst groups, but often groups of equal size are used to simplify the search. The groups may be pre-defined or chosen according to homegeneity. For this, define

$$\mathrm{SSE}(g_1, \ldots, g_J) = \sum_{j=1}^{J} \sum_{i \in g_j} (X_i - h(X_k, k \in g_j))^2,$$

where h is a function that determinates the replacement value (e.g. mean or median). The lower the SSE, the higher the within-group homogeneity.

Microaggregation

If X_i's are vectors with values in $\mathbb{R}^p$, we can calculate

$$\mathrm{SSE}(g_1, \ldots, g_J) = \sum_{j=1}^{J} \sum_{i \in g_j} \|X_i - h(X_k, k \in g_j)\|_2^2,$$

where $\|x\|_2 = \sqrt{x_1^2 + \cdots + x_p^2}$ is the Euclidean distance.
Alternatively, one can do microaggregation of each variable separately.

Microaggregation - example

A trivial example with pre-defined classes male/female using the mean function.

	age	gender	income	income_agg
1	50	Male	39641	36968.33
2	34	Male	39133	36968.33
3	33	Male	32131	36968.33
4	22	Female	74182	78842.67
5	56	Female	75179	78842.67
6	33	Female	87167	78842.67

Microaggregation - example

Note that the microaggregation here is applied to the sensitive attribute. Originally, without the microaggregation, we have 3-diversity if the attacker knows gender (we have 3 distinct values of income), while after the anonymization, we have 1-diversity - once you know the gender, you know the income. However, this is not the real income. If the attacker knows both age and gender, based on the original dataset we have SA disclosure (we know the income immediately). For the anonymized dataset we also guess income immediately, but this is not the true income.

```
   age   gender income income_agg
1   50     Male  39641 36968.33
2   34     Male  39133 36968.33
3   33     Male  32131 36968.33
```

Microaggregation

Technical comment: if you ask ChatGPT to divide set of size $n = 10$ into 5 groups of size 2, it will consider all possible permutations (how many of them we have? $n!$). Computationally, it becomes extremely costly very quickly.

Better ideas?

Technical aspects on finding the best groups are discussed here:

https://sdcpractice.readthedocs.io/en/latest/anon_methods.html

We will not discuss them here.

Noise addition

Noise addition, or noise masking, means adding or subtracting values to the original values of a variable, and is most suited to protect continuous variables.

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.

Noise addition

The simplest algorithm is to add consider

$$Y_j = X_j + \varepsilon_j \ , \ j = 1, \ldots, n,$$

where ε_j are i.i.d. $\mathcal{N}(0, \sigma^2)$, independent of X_j.

- Usually the mean should be zero to reduce bias of many statistics of interest: $\mathbb{E}[Y_j] = \mathbb{E}[X_j]$.
- Covariances are preserved: for $i \neq j$ we have

$$\begin{aligned}\mathrm{Cov}(Y_i, Y_j) &= \mathrm{Cov}(X_i + \varepsilon_i, X_j + \varepsilon_j) \\ &= \mathrm{Cov}(X_i, X_j) + \mathrm{Cov}(\varepsilon_i, \varepsilon_j) + \mathrm{Cov}(X_i, \varepsilon_j) + \mathrm{Cov}(\varepsilon_i, X_j) \\ &= \mathrm{Cov}(X_i, X_j) \ .\end{aligned}$$

Noise addition

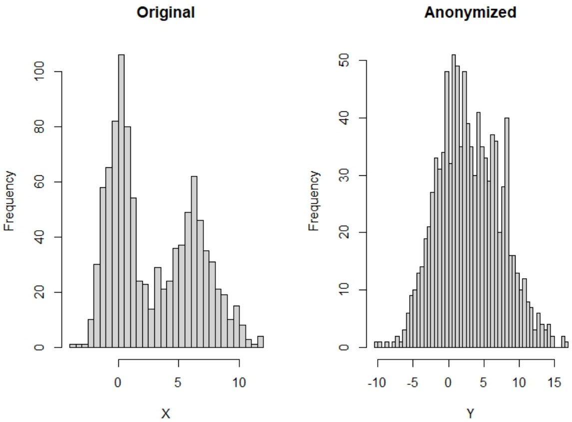
- Median, variance or correlation are not preserved:

$$\mathrm{Var}(Y_j) = \mathrm{Var}(X_j + \varepsilon_j) = \mathrm{Var}(X_j) + \mathrm{Var}(\varepsilon_j)\ .$$

- Note that these formulas make sense whenever we treat the data as random or as deterministic.
- Normal noise may not be theoretically justified (see *differential privacy*).

Noise addition

Too little noise fails to protect privacy. Too much noise may destroy the data structure. We will come back to this in section on data utility.



Noise addition - multivariate case

Assume our data $X_i = (X_{i1}, X_{i2})$, $i = 1, \ldots, n$. There is likely some dependence between $\{X_{i1}, i = 1, \ldots, n\}$ and $\{X_{i2}, i = 1, \ldots, n\}$. We can consider

$$Y_{i1} = X_{i1} + V_i \ , \quad Y_{i2} = X_{i2} + U_i \ ,$$

where U_i are i.i.d. random variables, V_i are i.i.d. random variables, both independent from the data. We can also assume that U_i's and V_i's are independent from each other. Note that the dependence structure of the anonymized data will be preserved. Often in SDC literature you can, however, find *if two or more variables are selected for noise addition, correlated noise addition is preferred to preserve the correlation structure in the data* ...