STATISTICAL DATA PROTECTION TECHNIQUES

Marta Más

EUSKAL ESTATISTIKA ERAKUNDEA
INSTITUTO VASCO DE ESTADISTICA

Donostia-San Sebastián, 1
01010 VITORIA-GASTEIZ
Tel.: 945 01 75 00
Fax.: 945 01 75 01
E-mail: eustat@eustat.es
www.eustat.es

Indice

Chapter

1

Introduction

Introduction and Objectives

The main objective of this technical notebook is the classification and description of the most important Statistical Data Protection techniques, which are commonly applied by the statistical agencies.

The need to control the enormous amount of data and information which flow through the international computer networks justifies the application of these control techniques before this data is published or disseminated. In this way any harm or prejudice is avoided to individuals or entities whose data could be accessed by anyone from any part of the world.

It is not our purpose to, in any way, veto the right to information but to balance it with the basic right to privacy. The aim is to produce the highest quality data with the safest guarantees.

Basic Security concepts

As in many areas of knowledge, Statistical Data Protection uses its own terminology, which always has to be defined in context. Therefore, it is necessary to clarify beforehand certain concepts that will occur repeatedly in this notebook.

Previous Definitions

- ***Data.*** This refers to representations of reality, the world, individuals and the possible relationships between them. Statistically talking, *data* refers to information organised for analysis and decision making.

- ***Confidential Data.*** This refers to *data* which cannot be published or disseminated due to ethical reasons, or previous mutual agreements with the survey respondents or the data owner.

- ***Sensitive Data.*** This refers to *data* which although not being confidential, does enable confidential information to be accurately estimated.(*)

- ***Safe Data.*** This refers to data which is non-confidential and, in addition, does not add any information about confidential data. We will talk about *safe* microdata files or *safe* tables depending on the case.

(*) Many times during the writing of this notebook, the terms "sensitive" and "confidential" are used in an interchangeable way, always referring to data which can not be published or the disclosure of which compromises confidential information.

- ***Intruder.*** Also called an *attacker* or *data spy*, this refers to the person/s who, intentionally or otherwise, individually or collectively, compromise sensitive information.

Privacy and Confidentiality

To understand the need for developing specific techniques to control statistical disclosure, we have to first acknowledge the right of the individual to privacy. By individual privacy, we mean *the freedom of the individual to decide how much of the self is to be revealed to others, when and to whom*. Privacy can be thought of as a state of the person and thus as a "personal property" right.

On the other hand, the concept of *confidentiality* reflects the desire of an individual *to restrict external access to personal information which may be used for specific aims or purposes*. In contrast with the above definition, confidentiality can be shown as a state of data.(*)

There is a definite relationship between confidentiality and privacy. Breach of confidentiality can result in a disclosure of data which harms the individual. This might be considered as an attack on privacy and therefore as a violation of a basic individual right.

The preservation of confidentiality is often required by law or government regulations which often vary from one country to another. Nevertheless confidentiality protection is also regarded as ethical behaviour in the statistics profession, thus it is a recognised responsibility of statistical agencies and institutes.

Identification and Disclosure

Disclosure is a difficult topic. To define what constitutes real disclosure is not a trivial issue and many authors disagree about it. Is it necessary to learn a sensitive attribute of a respondent or is it enough to identify a particular individual within a population? Is there a disclosure when a sensitive attribute is learned without previous identification?[19].

An *identification* occurs when a respondent is linked to a particular group of attributes (confidential or not) measured in a population or in a sample. On many occasions this is enough to compromise data integrity and question the security of the data system. However, recent trends consider only disclosures that involve identifications, thus, formalising a definition, a *disclosure* is said to take place when:

i) An identification occurs *(identity disclosure)*

ii) Confidential or sensitive information about the identified respondent is learned *(attribute disclosure)*

This definition leads to several types of disclosure classified by the information learned in every case. These are the most common:

- ***Exact Disclosure.*** This occurs when the exact value of the confidential attribute is learned and linked to the specific unit in the population.

- ***Statistical/Interval Disclosure***. This occurs when "too narrow" an interval or "too accurate" an estimation for the sensitive data may be deduced.

(*) Rieken, H.V. (1983) "Solutions to Ethical and Legal Problems in Social Research" New York: Academic press 1-9.

- ***True Disclosure.*** This occurs when the exact or statistical value of the obtained sensitive data, represents the real status of the attribute at the time of dissemination.

- ***Apparent Disclosure.*** This occurs when the exact or statistical value of the sensitive data, represents a plausible but not a valid or real value of the attribute at the time of dissemination.

- ***Positive or Internal Disclosure.*** This occurs when the information learned pertains directly to the identified unit.

- ***Negative or External Disclosure.*** This occurs when the information learned does not pertain to the identified unit but to its complements. That is, the attributes and characteristics learned do not belong to the unit itself but to others directly related to it. (e.g., competitor companies, professional colleagues,...).

All efforts will be centred on the avoidance of any kind of disclosure. Multiple access to information and the wide variety of data formats we can find nowadays make this no easy task. Restricted access to databases, by means of passwords or special permissions for authorised users, help prevent exact disclosures. However, more indirect attacks may take enabling information to be deduced from "inoffensive" data. This is called the *Inference Problem* [18].

The Inference Problem

As we have already explained in the previous section, a disclosure takes place when secret or confidential attributes are learned about an identified individual or unit in the population. If the professional secret directives and the confidentiality preservation rules are well applied and they guarantee the non-accessibility to confidential data to the general public, how can this confidential information be uncovered?.

How to avoid the inference or deduction of sensitive data from apparently "clean" information is one of the main problems to solve when implementing efficient data protection techniques. Some important factors to take into account when we search for solutions to the Inference Problem are the following ones:

- *Geographic Variables*. Geographical information could be of considerable help in localising small areas, within which it is easy to recognise individuals. Usually, direct regional indicators are not published and some statistical agencies and bureaux only provide geographic details when an area is big enough to avoid this kind of identification.

- *Sample Weights.* Often applied to correct sample defects, these might be able to give stratum information and thus, enable it to be attributed to an identified respondent belonging to the stratum. These weights refer in many cases to regional or geographic information which is data protector policy not to release. The disclosure of these characteristics places the agency in an embarrassing situation related to the preservation of confidentiality.

- *Available External Information*. Prior knowledge about the population, which may be held by any user, could help to deduce confidential information from initially safe data.

- The deduction of the *decision process* followed by an hypothetical intruder, who attempts to make a confidential inference, could be useful when implementing efficient data protection methods.

Several solutions to the inference problem based on these last factors are given. The *Bayesian approach* suits perfectly as a probabilistic solution when the prior knowledge is taken into account. However, to develop a working model for the intruder's behaviour is not an easy problem and requires more complex models.(*)

Sometimes, too complicated mathematical models generate less practical solutions or solutions of difficult implementation. Most of the techniques which are treated in this notebook will be based on the so called *heuristic methods.* That is, theoretic models exist to support convergence to an optimal solution (or demonstration that no solution exists), by means of finite algorithms (they end in a finite number of steps or iterations). The main drawback with these methods resides in time efficiency, which cannot be assured in all cases.

Disclosure Risk and Information Loss

Two basic concepts in the data protection universe are *disclosure risk* and *information loss*. The principal objective of data protection techniques is to provide maximum safety with minimum information loss. Quantitative and qualitative measures for both aspects will be given in order to compare the effectiveness between diverse methods and to study the repercussions on data quality and usefulness.

Data Protection Techniques

Data Protection Techniques should take into account all three steps of statistical data production*: data collection, data processing* and *data dissemination.* The principal studies have centred their efforts on the last phase of production. Therefore, most of these methods are called *Statistical Disclosure Control techniques.*

Distinctions are made between the ways in which data is released and technical details and methodological approaches differ depending upon the type of data product. Data is typically released in one of the three following ways:

- *Microdata files*.
- *Tabular data* (magnitude or frequency tables).
- *Sequential Queries to Databases*.

We will briefly describe the most commonly applied techniques for every dissemination format. Some of them will be shown in detail throughout this notebook.

Microdata Files Protection Techniques

A microdata file is a *unit record file which contains identifying information and other attributes pertaining to individual respondents in a sample or in a population*. The disclosure risk in a microdata file is mainly the *risk of identification*.

Direct identifying characteristics, such as name or address, are not usually published in external or publicly used files. However, attributes such as age, sex, marital status, profession..., are included and they are useful for identification. Thus, it is possible to link an individual respondent to a record on a microdata file. This matching could happen

(*) Duncan, G. and Lambert, D. (1986) "Disclosure Limited Data Dissemination" Journal of Business and Economic Statistics,81,10-18 ; (1989) "The Risk of Disclosure for Microdata" Journal of Business and Economic Statistics,7,207-217

because there may be individuals in the population who posses a unique combination of characteristic variables on the file. Consequently, it would be possible to learn all the information registered about the identified individual on the file, and any other data available on external files pertaining to the same sample or population.

The main purpose of microdata files protection techniques is to prevent identification of "rare" or unique individuals in the population with respect to a certain combination of characteristics, and to avoid the dissemination of unusual variable values which may aid this identification.

In general, we do not have available data for all population but only a significant sample. Therefore units or individuals which are "rare" in the sample do not have to be necessarily "rare" or unique in the population. An estimation of the *proportion of unique individuals in a population,* based on the proportion of individuals which are unique in the sample, could be a good measure of the effectiveness for the applied methods. That is, it could be a *quantitative measure of disclosure risk.*

These methods can be broadly classified in two principal groups:

- **Restriction Methods**. These have in common the fact that they all limit, by means of different techniques, the amount of information to be published:

    - *Global Recoding*: Several categories of a qualitative variable are collapsed into a single one, or extreme values in magnitude variables are grouped in top or bottom categories (top-bottom coding).

    - *Local Suppressions*: One or more sensitive values are suppressed for publication because of its high visibility (very unusual jobs, very large incomes,...).

- **Perturbation Methods.** These modify the values of certain variables allowing publication in a way that the exact values cannot be discovered. The following ones may be mentioned:

    - *Random Rounding.* The values of certain variables are replaced by quantities rounded to an integer base.

    - *Adding noise.* Small random numbers are added to the values of variables before they are published.

    - *Data Swapping.* This consists of interchanging attribute values between different records in such a way that the risk of identification decreases but certain summary statistics are preserved (e.g. the correlation structure is maintained).

    - *Microaggregation.* The data is ordered and partitioned into groups and all the attribute values are replaced in each record by the group values (mean for each group). More details about this method will be shown later in this notebook.

The ***cryptographic techniques*** are particularly worthy of mention. These have not been included above due to differences in the implementation and the final purpose of application. These techniques could be applied to any of the stages in the production process of data (*collection, processing, dissemination*) and they afford an efficient solution to the problem of *secure delegation* of data. That is, they permit the information to be processed by external agents in such a way that the real values of variables cannot be found out. Furthermore, these techniques permit a secure file transference between several information systems, limiting the danger of being intercepted and decrypted by

an intruder. A cryptographic technique will be developed in this notebook as a solution to the secure delegation problem.

Tabular Data Protection Techniques

Tabular data format (frequency count tables or magnitude tables) is a common way of releasing data. Therefore it is necessary to develop specific methods which provide safety without limiting information to the user.

Although data in tables is summarised and the aggregate level is higher than in microdata files, the risk of disclosure does not completely disappear. Small counts in frequency tables, and dominant contributions to a cell value in magnitude tables, could release sensitive information about the units or individuals which contribute to each cell.

Risk of disclosure in cells is given by *sensitivity rules* that will determine if a certain cell is confidential or not. The total number of sensitive cells in a table will quantify the disclosure risk of the whole table. The higher this number is, the lower the level of safety in the table.

Most protection methods for microdata files could be adapted to the tabular case. The next section summarises some of the principal ones:

- ***Controlled Rounding.*** Differs from *Random Rounding* in that application of the rounding in every cell has to be controlled, in order to maintain consistency between totals and subtotals within the table.

- ***Recoding.*** Consists in unifying categories in qualitative or quantitative variables to make the new cell value big enough to be considered safe.

- ***Granularity.*** This may be regarded as more of a safety criterion than a protection technique in itself. It is based on the unitary cell distribution (value 1 in frequency tables or unique contributions in magnitude tables) within the table. If the number of unitary cells is over a certain ratio or granularity index, the table is considered unsafe. Due to the parallelism between this technique and the identification of "rare" or unique individuals on microdata files, it may be possible to use some of the techniques outlined in this section to protect tables with high granularity levels.

- ***Cell Suppression.*** Local suppression of sensitive cells in tabular data is not enough to efficiently protect a table. Due to the totals, subtotals and other cell values it is easy to compute the value of a suppressed confidential cell. Therefore it is necessary to find a set of *secondary suppressions* to ensure that the values of the sensitive cells cannot be accurately estimated. To find an optimal suppression pattern, which provides the necessary safety with the minimum number of suppressions, is no easy matter. An efficient solution may be found linear programming methods which work well for low dimensional tables (3 or 4 dimensions).

These last two techniques will be developed in detail later in this notebook and some sensitive rules and information loss measures will be explained and applied depending on the case.

Database Protection Techniques

Databases provide a rich information environment in which not only data is stored but also the relationships and links between entries. Easy access to information by means of

sequential query systems leads to a high disclosure risk and puts in danger the database security.

Security in most databases is exclusively based on restricted access systems where only authorised users are able to use certain resources or work with specific information. However, there are multiple ways of inferring sensitive data. Information obtained from previous queries or from external databases, and the user's prior knowledge of the population might help to deduce confidential data. The solution to this inference problem in databases requires techniques which are capable of controlling on-line queries and decide if they can be answered in a secure way.

Two database protection methods which take into account the aforementioned aspects, are the following:

- ***Confidentiality via Camouflage (C.V.C.).*** Confidentiality is maintained by "hiding" the sensitive information in an infinite set of values contained between two bounds given as a response to the query.

- ***Auditing Databases.*** This consists of controlling the answers to sequential queries in on-line databases to avoid the inference of sensitive data due to data obtained from previous queries.

Statistical Disclosure Control Software

As well as analysing the different protection techniques, we will describe the software package ARGUS which has been specially designed for safe data production. It consists of two modules: μ-Argus to protect microdata files and τ-Argus to produce safe tables. Both have been developed at Statistics Netherlands as part of the latest European Statistical Disclosure Control project.

In addition, the program has been designed for an interactive environment (it works under Windows 95'-98') and it is easily used. It can also efficiently apply the most common protection tools found nowadays.

Application examples for both modules will be shown in the appropriate sections in this notebook, as well as some theoretical notes on the mathematical models they are based on.

Chapter
2

Database Protection

Confidentiality Via Camouflage

This practical method is presented by Gopal & Goes [12], focussed towards giving safe numerical response to database queries. The technique is appropriate for any size of database and no assumptions are needed about the statistical distribution of the confidential data. With this method any imaginable query type can be answered in a correct and unlimited way without compromising confidential numerical data.

Approaches to the Problem

Previous approaches to the problem of answering numerical queries while providing protection had been developed by means of several methods already used in microdata files. *(Perturbation and Restriction methods)*.

The Confidentiality Via Camouflage (C.V.C) seeks to incorporate the advantages of the above methods while eliminating their major disadvantages. As in perturbation methods, the C.V.C. is able to provide an unlimited number of answers and all of them are correct as in restriction techniques. The responses are in the form of a number plus a guarantee, so that the user can determine an interval which is sure to contain the exact answer. Confidentiality is maintained by "hiding" the vector of sensitive data in an infinite set of vectors. The advantages of this technique are clear:

- The technique is valid independent of the statistical distribution of data.
- Theoretically, any imaginable query type can be handled (unlimited response).
- There is no problem in dealing with static or dynamic databases of any size.
- The database manager can control which query types, based on the non-confidential fields, are likely to yield the tightest response intervals and therefore the closest to the exact answer.

The Model

The database can be considered to consist of n records each corresponding to a unit or an individual of the population stored in this database. The fields of the database are defined as confidential or non-confidential. For purposes of the analysis one confidential numerical field is considered. Therefore, let the numerical vector of confidential data be given by:

$$a = (a_1, a_2, \ldots\ldots\ldots, a_n)$$

The data manager, with free access to all database records, may associate either a finite lower bound (l_i), a finite upper bound (u_i) or both for each a_i . Then a_i is *protected* if no user can determine from the answered queries that:

$$l_i < a_i \quad \text{or} \quad u_i > a_i \quad \text{or} \quad l_i < a_i < u_i$$

Every query will refer to a set of records $T \subseteq N = \{1,2,.....n\}$ which satisfy some properties associated with one or more fields. If this query does not involve the confidential field it is called *benign.* When we refer to a query during the analysis we consider it as *non-benign*, that is, it implies a confidential field.

The response to a query will be a *point answer* (*r*) as well as a guarantee (*g*). Then if *e* is the exact correct answer it is true in every case that $e \in I = [r^-, r^+] = [r-g, r+g]$.

It is desirable to have *r* close to *e* with a small *g* and without the user being able to learn from the answers to the queries that $a_i \in (l_i,u_i)$. The final objective of the technique is to answer all queries with good, correct answers while providing protection..

The Technique

In the CVC technique, confidentiality is maintained by "hiding" the vector **a** in an infinite set of vectors P. All queries are answered as if they pertained to this infinite set and the user can never learn more about **a** from the answers to queries that $\mathbf{a} \in \mathrm{P}$. In addition, for each unit or individual i, at least one of the vectors of P always contains a number not exceeding l_i and another contains a number no less than u_i . Thus, the user can never determine that $a_i \in (l_i,u_i)$.

Let the set of vectors be the following:

$$P = \{P^1, P^2, ..., P^{k-1}, P^k\} \quad \text{with} \quad P^j = \{p_1^j, p_2^j, ..., p_n^j\} \quad \text{and} \quad P^k = a = (a_1, a_2, ..., a_n)$$

Let $p_i^- = \min_{j=1,....,k} p_i^j$ and $p_i^+ = \max_{j=1,....k} p_i^j$ which satisfy:

$p_i^- \leq l_i$ if l_i is specified (1)

$p_i^+ \geq u_i$ if u_i is specified (2)

The confidential vector **a** is "hiding" in the infinite set P = *conv(P).*

Index k represents the number of numerical fields of the database and in general is expected to be small ($3 \leq k \leq 6$) in most cases.

It is assumed that most numerical queries can be represented by single o multiple-valued functions f(**x**). Then $[r^-, r^+]$ can be written as:

$$r^- = r^l = \min f(\mathbf{x}), \quad \mathbf{x} \in \mathrm{P}$$

$$r^+ = r^u = \max f(\mathbf{x}), \quad \mathbf{x} \in \mathrm{P}$$

Or equivalently :

$$r^{l} = \min f(l) = \min f(\sum_{j=1}^{k} l_j \cdot P^j), \quad \sum_{j=1}^{k} l_j = 1 \quad l_j \geq 0, \quad j = 1,.....,k \tag{3}$$

$$r^{u} = \max f(l) = \max f(\sum_{j=1}^{k} l_j \cdot P^j), \quad \sum_{j=1}^{k} l_j = 1 \quad l_j \geq 0, \quad j = 1,.....,k \tag{4}$$

Where $l = (l_1, l_2,....., l_k)$. It follows that **e** ∈ $[r^{l}, r^{u}]$ since **a** ∈ P. Protection is guaranteed by (1) and (2).

For some queries the expressions (3) and (4) can be solved easily (e.g. when f(l) is linear so that there exists an optimal l which are de k unit vectors), so that answering $[r^{l}, r^{u}]$ is computationally feasible. For others that may not be the case. Then we answer with $[r^{-}, r^{+}]$ where $r^{-} \leq r^{l}$ y $r^{+} \geq r^{u}$ to ensure protection. The key will be to efficiently calculate these values which give the smallest guarantees.

For a given query, each record has to be accessed no more frequently than if the answer were f(**a**) (that is , in terms of the confidential data). This last consideration is the key to the efficiency of the method.

Audits for Databases

In an on-line database environment, it is necessary to keep certain control over the queries that a potential user could make. Not only because of the queries which concern confidential data but also those which involve sensitive data which, when added to other information obtained in previous queries, could give some "clues" to the exact value of the confidential attribute. An audit for databases takes into account this matter avoiding the inference of sensitive information and using a mathematical model in which the number of variables is never greater (usually far less) than the size of the underlying database [20].

Previous Concepts

Before a full audit implementation, it is necessary to familiarise ourselves with the specific terminology which is going to be used later. For a given query we'll talk about:

- **Logical Formula**

This is also called *categorical* or *characteristic* and it is built up from assigned values for certain attributes (field names) by means of Boolean and conditional operators ($\wedge, \vee, \neg$) and selects a set of records from the database known as the **query-set** of the query.

- **Aggregation Function**

This takes as its inputs a numeric attribute and it returns a summary *value* as a result of evaluating the expression (SUM, COUNT, AVERAGE, MAX, MIN,...) in the *query-set*.

(See Example)

Sensitivity Rules

When the query-processing system recognises a statistical query as being sensitive (the knowledge of its value may lead to an accurate estimate of a confidential attribute), it will deny the answer. The exact rule for defining sensitive queries is determined by the data type and the database system but the following are the most common used:

- **Threshold Rule (for count-queries)**

Sensitivity is determined by an appropriately chosen positive integer n, and defines a count-query to be sensitive if its query-set contains n or fewer records. This value n is chosen so that the probability of identifying any record, as a member of an arbitrary set of n+1 or more records is acceptably small with respect to established criteria.

- **Dominance Rule (N, K) (for sum-queries)**

It defines a sum-query q to be sensitive if N or fewer records in the query-set constitute more than K% of the total summary value.

Inferability from a set of queries

Even if a query is not sensitive, answering all type of queries limiting the protection to the sensitivity rules, endangers the safety of confidential information since the user might infer some sensitive summary data on the basis of the answers to previous queries. Therefore, we will talk about *safe* sets of queries and inference methods which will decide if answering a new query leads to a disclosure of confidential data.

Definition 1. Given a sequential set of answered queries $\{q_1, q_2, \ldots q_n\}$ that it is considered *safe* , a new query q_{n+1} will decide if q_{n+1} can be answered safely.

Notation:

Let Q be a set of **sum-queries** and S_q the *query-set* of $q \in Q$. Let R be the union of all the sets S_q and t_q the summary value for every q.

A unique partition π of R is created formed by disjoint classes in such a way that each non-empty S_q is the union of one or more classes of π. The values of queries t_q will be summarised in a linear constraint system, the variables of which correspond 1-1 to classes of π. The classes of π are exactly the non-empty values of the $2^{|Q|}$ -1 set expressions of the type:

$\bigcap_{q \in Q} s_q$ where s_q is either S_q or $\overline{S}_q = R - S_q$ with the exclusion of the set

$\bigcap_{q \in Q} \overline{S}_q$

By the *query diagram* for Q we mean the hypergraph H with vertex set Q and edge set given by:

$$E = \{e \subseteq Q \mid (\bigcap_{q \in e} S_q) \bigcap (\bigcap_{q \notin e} \overline{S}_q) \text{ is a class of } p\}$$

For each edge e of H, we denote by x(e) the class variable corresponding to e and we construct the linear constraint system **Mx=t**, where M is the vertex-edge matrix of H.

Let A be a subset of the edge set E. We consider the following sum expression of which A is called the *support* : $x(A) = \sum_{e \in A} x(e)$

We say that x(A) is an *invariant* of the linear system if x(A) is constant, that is, if for every two system solutions $\mathbf{x_1}$ y $\mathbf{x_2}$ we have:

$$\sum_{e \in A} x_1(e) = \sum_{e \in A} x_2(e)$$

Definition 2. An edge set of H is an *invariant-set* if it is the support of an invariant of the system.

Definition 3. Given a set of records S, let X be the summary data obtained by evaluating the aggregation function in all records in S. We say that X is *inferable* from Q if either S=∅, or there exists a non-empty subset A, of the edge set of the query diagram for Q, such that:

$(i) \quad S = \bigcup_{e \in A}[(\bigcap_{q \in e} S_q) \bigcap (\bigcap_{q \notin e} \overline{S}_q) \quad (Covering \;\; property)$

(ii) A is an invariant set of H $\qquad$ *(Invariance property)*

Thus, we could summarise these two properties saying that a set of sequential queries is considered safe, if the information (X), inferred from the linear constraint system constructed from this set of queries, is not sensitive.

An algorithm based on the above rules will let us decide if a certain query can be answered safely, without implying the inference of sensitive data.

The Answerability Test

This test will help us to decide if, for a given safe set of answered queries $\{q_1, q_2, \ldots q_n\}$, the response to a new query q_{n+1} can be made in a safe manner. In addition, it will determine if the new set of queries $\{q_1, q_2, \ldots q_n, q_{n+1}\}$ is also safe.

The test will take as input a subset Q of $\{q_1, q_2, \ldots q_n\}$, called the *basis,* for which the value of any $q_i \notin Q$ is inferable from $Q \cap \{q_1, q_2, \ldots q_{i-1}\}$ and the inferred information is not sensitive. For a given non-sensitive q_{n+1} satisfying this property, Q will not vary and we will answer q_{n+1}. Additionally, $\{q_1, q_2, \ldots q_n, q_{n+1}\}$ will be considered safe. If this is not the case, there might exist a sensitive edge subset of E that added to the q_{n+1} value, could disseminate sensitive information. To detect this circumstance, we will create a new *basis* Q' for the new set $\{q_1, q_2, \ldots q_n, q_{n+1}\}$ and we will apply a *security test* to it, checking each edge by means of the sensitive test. If the result is negative, ($\{q_1, q_2, \ldots q_n, q_{n+1}\}$ is not safe) q_{n+1} will not be answered. In the affirmative, q_{n+1} will be released and Q' will be the new *basis* of reference for further queries.

The steps followed by the algorithm are summarised as follows:

Input: q_{n+1} ; H=(Q,E) is the query-diagram of Q with E = edge set of Q.

Step 1. *Sensitivity Test* to q_{n+1}

If q_{n+1} is sensitive then deny the answer and **Exit.**

Otherwise, go to Step 2.

Step 2. *Inferability Test*

If H satisfies (*covering AND invariance property)* then q_{n+1} is inferable from Q. q_{n+1} will be released and the *basis* of $\{q_1, q_2, \ldots q_n, q_{n+1}\}$ will be Q again. **Exit.**

Otherwise Step 3.

Step 3. A new query diagram H' for $Q \cup \{q_{n+1}\}$ is constructed.

Step 4. *Safety Test* to H'

If $Q \cup \{q_{n+1}\}$ is not safe then deny the answer and **Exit.**

Otherwise q_{n+1} is released and $Q' = Q \cup \{q_{n+1}\}$ will be the new *basis* for next queries.

Although this algorithm is provided, it is still necessary to optimise the sensitive edge detection process (Safety Test) [*] in order to apply it to other cases, which are different from the real data.

Example

We shall identify in the next example proposed in [20] some of the described concepts:

Consider the following database EMP containing 9 records of people working in a hypothetical company. Assume that each employee is described by several attributes such as ID (the employee code), DEP (the department where the employee works) and SALARY (the employee's salary).

Record	ID	DEP	SALARY
1	id1	Management	85
2	id2	Management	3
3	id3	Management	2
4	id4	Administration	4
5	id5	Administration	3
6	id6	Administration	3
7	id7	Services	4
8	id8	Services	3
9	id9	Services	3

The following SQL-queries are simple examples of sum-queries having SALARY as their summary attribute:

***q_1**: select SUM(SALARY) from EMPRESA where DEP=Administration*

Value of q_1 = t_1 =10

***q_2**:select SUM(SALARY) from EMPRESA where DEP=Servicios*

Value of q_2= t_2 =10

***q_3**:select SUM(SALARY) from EMPRESA*

Value of q_3= t_3 =110

***q_4**:select SUM(SALARY) from EMPRESA where DEP=Management*

Value of q_4= t_4 =90

According to the sensitivity criterion specified by the (N, k)=(2, 85%) dominance rule, q_4 is sensitive because there exists one contribution (record 1) which supposes more than the 85% of the query total value. The other queries are not sensitive with respect to this criterion.

We are going to identify the logical and aggregation formulas for each non-sensitive q_i:

- *Aggregation Function*: SUM(SALARY) in all cases.

[*] More information about the Safety Test could be found in Malvestuto,F.M., Moscarini, M. (1990) "Query evaluability in statistical databases" IEEE Transactions on knowledge and data engineering 2, 425-430.

- *Logical Formulas*	*Query-set*
q₁: DEP=Administration	S_1={4,5,6}
q₂: DEP=Services	S_2={7,8,9}
q₃: True	S_3={1,2,3,.....,9}

Q and R sets, partition π and the edge set E, are determined by:

$Q=\{q_1,q_2,q_3\}$ $\quad R=\bigcup S_i = S_3$

Partition classes are given by the following expressions (the non-empty ones of the 2^3-1 possible expressions):

$$S_1 \cap \overline{S}_2 \cap S_3 = S_1 = \{4,5,6\}$$

$$\overline{S}_1 \cap S_2 \cap S_3 = S_2 = \{7,8,9\}$$

$$\overline{S}_1 \cap \overline{S}_2 \cap S_3 = S_3 - (S_1 \cup S_2) = \{1,2,3\}$$

Edge-set $E=\{e_1,e_2,e_3\}=\{ (q_1,q_3), (q_2,q_3), (q_3) \}$ with:

$$For\ e_1 : (S_1 \cap S_3) \cap \overline{S}_2 = \{4,5,6\}\ is\ a\ class\ of\ p$$

$$For\ e_2 : (S_2 \cap S_3) \cap \overline{S}_1 = \{7,8,9\}\ is\ a\ class\ of\ p$$

$$For\ e_1 : S_3 \cap (\overline{S}_1 \cap \overline{S}_2) = \{1,2,3\}\ is\ a\ class\ of\ p$$

Let x_i be the variable which is associated to each class of π. The linear constraint system Mx=t, which summarises the information provided by the answers to queries q_1,q_2 and q_3, is the following:

$$\begin{cases} x_1 = 10 \\ x_2 = 10 \\ x_1 + x_2 + x_3 = 110 \end{cases} \quad \text{with } x_i \geq 0$$

The value of x_3, which is the addition of all salaries in the management department, is uniquely determined by the system (x_3 = 90). For this reason and for being a sensitive value according to the applied sensitive rule, the real value of the id1 employee's salary can be accurately estimated.

Finally, we conclude that Q is not a safe set of queries because sensitive information can be inferred from given responses. It would be recommendable to apply the answerability test before answering q_3, that although not being sensitive, it does provide information about confidential data in the database.

Chapter

3

Microdata File Protection

Microaggregation

Microaggregation is a statistical disclosure control technique for microdata. Individual records in the file are grouped into small aggregates prior to publication. Until now, microaggregation has consisted of taking fixed-size microaggregates (size k). In this section we consider some of these methods along with an approach to multivariate microaggregation in which the size of aggregates is a variable taking values ($\geq$ k) depending on data distribution [4].

Approach to the problem

The rationale behind microaggregation is that there exist certain confidentiality rules which permit the publication of microdata sets along as the data vectors correspond to groups of k or more individuals, where no individual dominates (i.e. contributes too much to) the group. Strict application of such confidentiality rules leads to replacing individual values computed on small aggregates (means for each group).

To obtain these aggregates in a microdata set with n data vectors, these vectors are combined to form g groups of size at least k. For each numerical variable the average value of each group is computed and is used to replace each of the original values.

Notation:

We consider a microdata set with **p** continuous variables and **n** data vectors (e.g. the result of observing p variables on n individuals). With these individuals **g** groups are formed with $\mathbf{n}_i$ individuals in thei-th group ($n_i \geq k$ and $n = \Sigma n_i$).

We denote:

$\mathbf{X}'=(x_1,x_2,.....,x_p)$ data vector where x_i are variables values.

$\mathbf{X}_{ij}$ j-th data vector in the i-th group $j=1,.....,n_i$

$\bar{\mathbf{X}}_i$ average data vector of the i-th group $i=1,.....,g$

$\bar{\mathbf{X}}$ average data vector over the whole set of data

To solve the k-partition problem, a measure of similarity between data vectors is needed. Each individual data vector can be viewed as a point and the whole microdata set as a set of multidimensional points (the dimension is the number of variables in data vectors). If data vectors are characterised as points, similarity between them can be measured using a *distance*:

Within-groups Sum of Squares

$$SSE = \sum_{i=1}^{g}\sum_{j=1}^{n_i}(\mathbf{x_{ij}} - \bar{\mathbf{x}}_\mathbf{i})'(\mathbf{x_{ij}} - \bar{\mathbf{x}}_\mathbf{i})$$

Between-groups Sum of Squares

$$SSA = \sum_{i=1}^{g} n_i(\bar{\mathbf{x}}_\mathbf{i} - \bar{\mathbf{x}})'(\bar{\mathbf{x}}_\mathbf{i} - \bar{\mathbf{x}})$$

Total Sum of Squares

$$SST = \sum_{i=1}^{g}\sum_{j=1}^{n_i}(\mathbf{x_{ij}} - \bar{\mathbf{x}})'(\mathbf{x_{ij}} - \bar{\mathbf{x}})$$

The optimal k-partition is the one that minimises SSE or equivalently, maximises SSA. Sums of squares are usual to *measure information loss.* A measure L of information loss standardised between 0 and 1 can be obtained from:

$$L = \frac{SSE}{SST} \qquad (1)$$

The closer to 0 L is, the lower the information loss is the due to microaggregation. The following heuristic methods form a practical unidimensional alternative which aims to minimise the information loss through the observation of data variability (SST *and* SSE).

Unidimensional Fixed-size Methods

These heuristic methods sort in ascending or descending order the data vectors according to a particular unidimensional criterion. Then, groups of successive *fixed-k* vectors are formed. If the total number of data vectors n is not a multiple of k, then the last group will contain more than k data vectors. Inside each group, the effect for each variable is to replace the k values taken by the variable, with their average.

- **Single-Axis sorting methods**

They are good if all variables are highly correlated. The data vectors can be sorted (ascending or descending) by means of several criteria:

- *Principal Components sorting.* Data vectors are sorted by the first principal component of the microdata set. Principal Components are transformed variables so that the first principal component is highly correlated with most of the original variables

- *Particular Variable sorting*. Vectors are sorted in ascending or descending order by the sorting variable, then this variable must reflect somehow the size of the data vector.

- *Sum of z-scores sorting*. All variables are standardised and, for each data vector, the standardised values of all variables are added. Vectors are subsequently sorted by their sum of z-scores.

- **Individual Sorting**

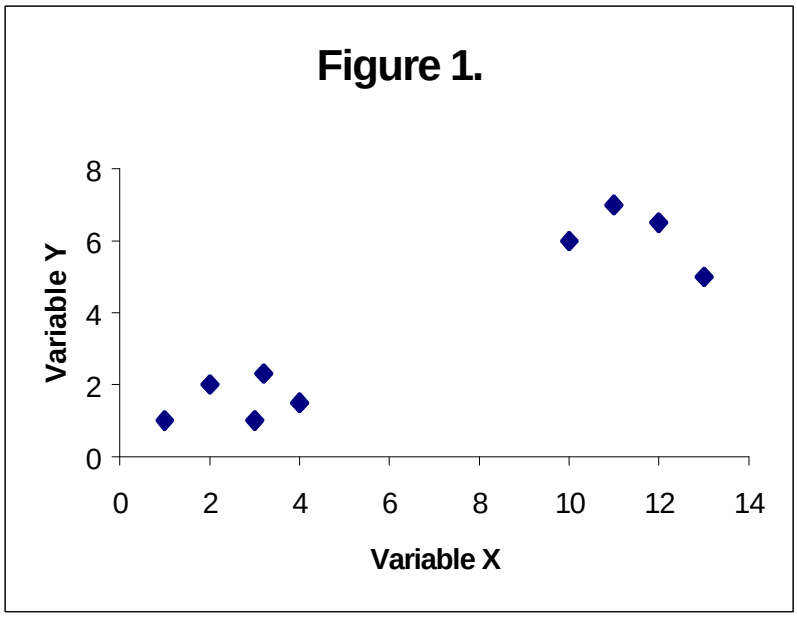
Each variable is considered independently. Data vectors are sorted by the first variable, then groups of k successive values of the first variable are formed and, inside each group, values are replaced by the group average. A similar procedure is repeated for the rest of variables (sorting by the second variable, third variable,...successively).

Individual sorting usually preserves more information than single-axis methods, but has a higher disclosure risk. Indeed, with individual sorting any intruder knows that the real value of a variable in a data vector in the i-th group is between the average of the i-1 th group and the average of the i+1 th group. If these two averages are very close to each other, then a very narrow interval for the real value being searched has been determined.

Individual sorting also has a conceptual drawback: instead of partitioning the n data vectors on a data vector basis, microaggregation is carried out for each variable, therefore a different partition (thus a different average vector) is obtained for each variable in the data set.

Variable-sized Methods

Microaggregation techniques most commonly used at the moment are fixed-size ones. However the use of variable-sized groups ($\geq$ k) depending on the initial data distribution, would tend to reduce information loss. The next figure shows the advantages of variable-sized microaggregation:



The figure shows two variables and nine data items. If fixed-sized microaggregation with k=3 is used, we obtain a partition of the data into three groups, which looks rather unnatural for the data distribution given. On the other hand, if variable-sized groups are allowed then the five data items on the left can be kept in a single group and the four data items on the right in another group. Such a variable-sized grouping achieves smaller information loss.

Two alternative heuristic approaches to variable-sized microaggregation are presented below:

- **Genetic Microaggregation**

This represents k-partitions as binary strings (also called chromosomes) and combines direct and random searches to locate the optimal partition. Unfortunately, such a genetic approach is not so easy to adapt to the multivariate case. The main problem comes from the fact that a multidimensional space is only partially ordered, which makes this binary representation far from appropriate.

- **Ward's Method**[(*)]

This is a fairly effective method which provides a recursive algorithm which optimises the solution in every step. The two groups or data elements joined at each step are chosen so that the increase in the within-groups sum of squares (SSE) caused by their union is minimal.

Ward's method was originally applied as a hierarchical classification method which was stepwise optimal but without considering any minimum size for the aggregates formed. Therefore, it has had to be adapted to be useful for microaggregation and is now called *Modified Ward's Algorithm*.

Modified Ward's Algorithm (MWA)

MWA is a microaggregation method for quantitative or qualitative data in which a distance has been defined. Following, we will briefly recall the steps followed by the algorithm for the *univariate case* [(**)]. The multivariate case will cause few changes in the algorithm which will be shown in detail in this section.

The following definitions and results are needed:

Definition 1. *For a given data set, a k-partition P is any partition of the data set such that each group in P consists of at least k elements.*

Definition 2. *For a given data set, k-partition P is said to be **"finer than"** k-partition P' if every group in P is contained by a group in P'.*

Definition 3. *For a given data set, a k-partition P is said to be **minimal** with respect to the relationship "finer than" if there is no k-partition P'=P such that P' is finer than P.*

Proposition 4. *For a given data set, k-partition P is minimal with respect to the relationship "finer than" if and only if consists of groups with sizes ³ k and < 2k.*

Corollary 5. *An optimal solution to the k-partition problem of a set of data exists that is minimal with respect to the relationship "finer than".*

(*) Ward,J.H.(1963) "Hierarchical grouping to optimize an objective function" *Journal of the American Statistical Association, 58, 236-244*

(**) For a more detailed explication see Domingo,J., Mateo, J.M. (1997) "Practical Data-Oriented Microaggregation for Statistical Disclosure Control" *Journal of Classification*

Algorithm. Univariate case

Step 1. *From an **ordered** data set two groups are formed: one with the first (smallest) k elements and the other with the last (largest) k elements of the data set.*

Step 2. *Use Ward's method until all the elements in the data set belong to a group containing k or more data elements; in the process of forming groups by Ward's method, never join two groups which have both a size greater than or equal to k..*

Step 3. *For each group in the final partition that contains 2k ore more data elements, apply this algorithm recursively.*

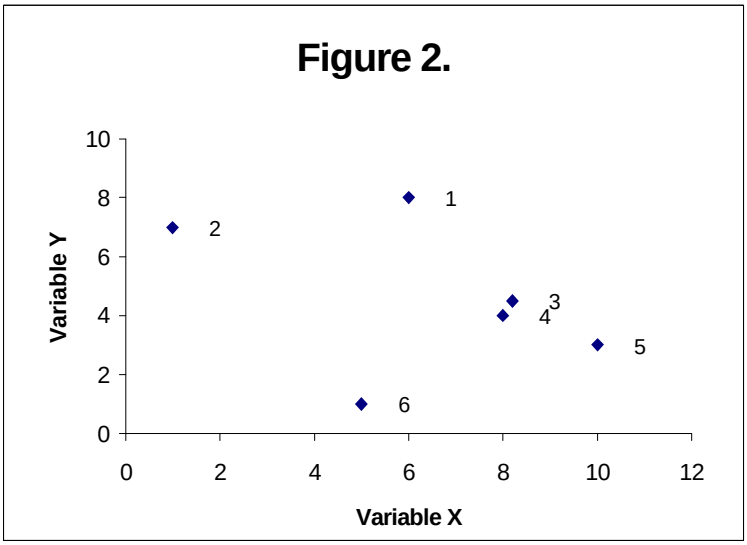
It may be demonstrated that the algorithm ends after a finite number of recursion steps and it obtains the minimal partition for the given k. Even though the optimality is not maintained on every step (as occurs in Ward's method), good performance is expected in terms of information loss.

Algorithm .Multivariate case.

The above algorithm can be easily adapted into the multivariate case by changing step 1. It is necessary to define clearly what is meant by the "first" k data elements and the "last" k data elements, as we are talking about multidimensional vectors.

Rather than using the single-axis or individual sorting to perform microaggregation itself, such procedures can be used as a vector sorting criteria in step 1 of the MWA.

However, there is an additional sorting criterion, which is specific for the multivariate case, this is called *Maximum-Distance Criterion.* It defines as extreme data vectors the two which are most distant according to the distance matrix. In this way, a group with the "first" k data vectors and another group with the "last" k data vectors are obtained. The resultant grouping may depend on which extreme data vector is taken as the "first" or "last" one. In Figure 2. this last consideration clearly shown:



If we consider the six two-dimensional data vectors and take k=3, the two more distant vectors are labelled 2 and 5. Starting from vector 2, the closest vector is 1; now the vector closest to group (1,2) is vector 3. Thus, we get the groups (1,2,3) and (4,5,6). However, if we choose to start from the other extreme point (vector 5), now we clearly get the groups (3,4,5) and (1,2,6). Nevertheless, the differences in the information loss resulting from choosing either extreme vector as "first" or "last" are small.

One of the disadvantages of using this sorting criterion is mainly related to storage capacity. The required distance matrix containing the distance between each pair of data is symmetrical and has zeroes on its diagonal. So, the storage needed for such a matrix and for a data set of size n is a quadratic function of n: (n-1)n/2.

On the other hand and in conclusion, it can be assured that the information loss resulting from the application of the MWA is usually smaller than the information loss resulting from the application of the other heuristic methods of microaggregation. Future research should tend to ameliorate the computational results both in time efficiency and in storage capacity.

Cryptography

On many occasions, statistical data has to be processed out of its place of origin. Therefore, it has to be delegated to an external party. If the data owner distrusts the external party or the data is confidential, delegation should be performed in a way that preserves security.

A legal solution to this problem is to require that the external party (or *handler*) signs a non-disclosure agreement. However, legal questions aside, the data owner must trust the handler, since there is no technical means to prevent the inadequate use of data.

In this section, we describe an implementation that allows the data owner to control and limit the kind of operations performed by the handler. This technique constitutes a cryptographic solution to the *secure delegation* problem [6].

Secure Delegation of Data

Two requirements must be met by secure delegation:

- **Data Secrecy**

The data owner must be assured that data remain secret. In order to compute the input data without revealing the original information, the processing can be carried out on previously *encrypted* data. The data owner must apply some encryption transformations, known as *Privacy Homomorphims*, which allow basic arithmetic operations (addition, subtraction, multiplication, fraction division,....) and even equality tests, to be carried out on encrypted data without breaching confidentiality.

- **Computation Verifiability**

The data owner must be able to verify that the computation carried out by the handler is correct.

One procedure to carry out the computation verifiability, without repeating the full computation process done by the handler, consists of a probabilistic verification called *parity checking.* The idea is to transform arithmetic operations into Boolean expressions on data parities, obtaining for each expression its *parity* (to compute if it is odd or even). If we denote Z(x) as the parity function, the parity expressions for a+b and a∗b could be expressed as follows:

$$Z(a+b)= Z(a) \oplus Z(b)=Z(a) \textbf{ OR } Z(b)=Z(o)$$

$$Z(a*b)= Z(a) \cdot Z(b) = Z(a) \textbf{ AND } Z(b)=Z(y)$$

Let the estimated parity distribution for an input datum d be $(1-p_d,p_d)$ where p_d is the probability of d being odd. The parity distributions for sum and product results are respectively:

$$Z(o) \text{ distribution is: } (1-p_o,p_o) \quad \text{con } p_o=p_a*(1-p_b) + (1-p_a)*p_b$$

$$Z(y) \text{ distribution is: } (1-p_y,p_y) \quad \text{con } p_y=p_a*p_b$$

Thus, it is possible to compute the parity of any Boolean expression (with AND/OR terms). Therefore, upon completing the computation on encrypted data, the handler returns the encrypted result with the *claimed expression* (the expression that the handler is supposed to have used in the processing of data). On one hand, the data owner decrypts the result (*computed expression*) and obtains its parity. On the other hand, the data owner computes the parity resulting from feeding the input data to the claimed expression. If the compute process has been correctly done, both parities have to be equal. If they differ, a fraud has been detected.

As has been demonstated, the parity distributions of the required and computed expressions are respectively (1-p,p) and (1-p',p') thus the *probability of fraud detection* is given by the following expression:

P(detect) = P(odd required expression) x P(even computed expression) +

+ P(even required expression) x P(odd computed expression)

Lemma. *If the claimed and the computed expressions differ, the probability of fraud detection by the owner is:*

P(detect)=p(1-p') + p'(1-p) with p = probability of the claimed expression being odd (1)

and p' = probability of the computed expression being odd

Theorem 1. *For random input data, the probability of fraud detection by parity checking of the final result is tightly upper-bounded by 1/2.*

Dem/ If input data are random, they have parity distributions (1/2,1/2). Additions introduce no parity bias but results of multiplication tend to be even (only odd x odd = odd). This means $p \leq 1/2$ and $p' \leq 1/2$ in (1), thus $P(detect) \leq 1/2$. If required and computed expressions contain only additions then p = p'= P(detect) = 1/2.

Theorem 2. *For random input data, the probability of fraud detection by parity checking is tightly lower bounded by max(p,p'). If always-even claimed expressions are forbidden by the data owner, then the probability of fraud detection is always > 0.*

To recognise an always-even expression, the data owner must check if the associated Boolean parity formula is equivalent to 0. An easy way of carrying out this check is requiring the claimed expression to be in sum-of-products (SOP) form. This way, it is enough to check for each product (AND terms) in the Boolean expression to see if its complement exists.

Finally, we conclude that the data owner can be assured of a nonzero detection probability but will be unable to compute the lower bound mentioned in Theorem 2. This is because the data owner does not know the value of p' (the real expression computed by the handler, if there is a fraud is not known). One way of increasing the fraud detection probability is to apply the parity checking to intermediate results. The more verified intermediate expressions the higher will be the probability of fraud detection, even though this might suppose an additional time cost for the data owner.

μ-Argus for Statistical Disclosure Control in Microfiles

The Objective

Producing safe micro data in the dissemination phase of data production is one of the main objectives of statistical agencies and institutes. To determine when a data set may be considered safe is not a trivial issue and requires previously determined safety criteria which promote a balance between the preservation of confidentiality rules and the growing demands for information on behalf of the user.

If a data set is not safe, according to the safety criteria imposed, the information will have to be modified in such a way that, the risk of disclosure is minimised and the maximum amount of information is preserved.

μ-Argus is a specific software designed for safe file production. It uses its own model and safety criteria which will be explained in this section[22]. Additionally, we will develop an application example of the software, using data obtained by EUSTAT from the Labour Force Survey. An explanation of the working and results obtained will also be provided.

The Model

μ-Argus is based on a *re-identification* model where a hypothetical intruder is able to identify a certain unit or individual in the population by linking up some attribute values, present in the microfile, with the mentioned unit or individual. The most identifying variables play an important role in this model. They will be called *key variables*. The possible key variable combinations, which are able to identify uniquely any individual or unit in the population, are called *identifying keys*.

Therefore, the aim is to determine which identifying keys lead to a higher risk of disclosure of individual information and to avoid, by means of *recodings* or *local suppressions* of certain variable values, the identification of "rare" or unique individuals in the population.

To be "rare" with respect to a certain key means that this combination of identifying values appears less than a certain threshold value denoted by D_k, where k represents the key. If this combination occurs more than D_k times in the population, then it is considered safe. Otherwise it will be considered unsafe and it will be necessary to apply the adequate modifications on data to protect against identification.

The threshold value D_k is determined by the data protector who is assumed to have a prior knowledge of the population and who knows what could be considered "rare" or not. In case of having sample data instead of the whole population data this threshold value is calculated at the level of the sample. To translate this at the level of the population the sample fraction should be taken into account.

Protection Techniques

Once the unsafe keys have been identified it is time to apply the adequate techniques on the data before they are published.

μ-Argus offers several disclosure control methods:

- *Global Recoding*. This is applied to all records in the microfile and consists in the aggregation of categories.

- *Local Suppression* of variable values in certain records the publication of which could results in an identity disclosure.

- *Perturbation methods* applied to numerical variables, they modify values in order to publish them, preventing their real or exact value to be found out (microaggregation, adding noise,...).

Often the application of one of those techniques is not enough to produce a safe file. To combine global recoding with local suppression (minimising the number of suppressions), could provide an optimal result in terms of information loss and disclosure risk.

Other Important Factors

Information Loss Measures

μ-Argus uses different quantifiers to evaluate information loss depending on the applied protection techniques. In the case of applying local suppression only, μ-Argus simply counts the number of suppressions carried out. The more suppressions, the higher the information loss. In case of global recoding μ-Argus uses an information loss measure based on a valuation of the importance of the identifying variables, as well as a valuation of each predefined coding for each identifying variable (the program permits interactive indication of these two parameters). This values are only needed in case we choose the available automatic option to produce safe data. This process could be carried out manually by the data protector in which case measures of information loss would be cruder and intuitive and they would be chosen depending on the type of data or their finality.

Sampling Weights

In μ-Argus it is possible to add noise to the weight variables before publication. As we have seen in Chapter 1 in this notebook, these variables could add information about the strata in the population and the characteristics which belong to them. This could lead to a high risk of disclosure.

Household Variables

The sampling of members in households is commonly applied to obtain representative samples of a population. Some identifiers necessarily yield the same scores for all members in the same household. Such variables are called household variables (e.g. "number of members in the household", "occupation of the head of the household",...). There should be a variable which identifies uniquely each household and it is not usually published. However, this identifying variable is used by μ-Argus to identify members of the same household, thus if certain household variable values are suppressed, then the corresponding values are suppressed in the records referring to the others members of the same household.

Regional Indicators

Many times, the dissemination of regional attributes (e.g. "degree of urbanisation", "number of inhabitants", "place where a person works",...) in individual records, can be of

much help in localising individuals within geographical areas. If these areas are too small the risk of disclosure rises and then the publication of this regional indicators are not recommended. μ-Argus, enables us to work with regional indicators and checks if small areas emerge from their intersections.

Example

We're going to use for this application example, the Labour Force Survey made in EUSTAT for the Basque Country. We consider a sample of 2,913 individuals belonging to the province of Alava. This sample corresponds to the three first months of 1996.

We describe the variables we are going to use during the analysis:

Eciv: Categorical variable which represents the marital status of the individual. The identification level assigned to this variable is 3 and is specified as follows:

0, NSP (Non-Sampling Population)
1, Single
2, Married
3, Widow/er
4, Divorced or Separate

Edad: This represents the age of the individual. The identification level assigned to this variable is 1. That means that it is a very identifying variable, partly due to the high number of categories.

Nivel: Categorical variable which represents the type of studies carried out by the individual. The identification level assigned to this variable is 3 and it is specified as follows:

0, NSP (Non-Sampling Population)
1, Primary studies
2, Secondary studies
3, University studies

Sexo: Categorical variable which represents the sex of the individual. The identification level assigned to this variable is 2. That means that it is a very identifying variable though the number of categories is not very large. It is specified as follows:

1, Man
2, Woman

Prof2: Categorical variable which represents the profession of the individual. The identification level assigned to this variable is 2 and it is specified as follows:

0, NSP
1, Superior Technicians and Professionals
2, Medium Technicians and Professionals
3, Managers
4, Administrative managers
5, Administrative personnel
6, Merchants and Salespersons
7, Administrative Auxiliaries
8, Other personnel of services

9, Farmers
10, Iron, Steel and Metallurgic workers
11, Forger workers and Tool makers
12, Mechanics and Precision Fitter
13, Electricians
14, Plumbers, Welders and Tinsmiths
15, Masons and Other workers of Construction
16, Drivers and Other workers of Transport
17, Other industrial workers

Busq1: Categorical variable which represents the situation of the individual in respect of the seeking of employment. The identification level assigned to this variable is 0 and it is specified as follows:

0, NSP (Non-Sampling Population)
1, Seeking employment
2, Not seeking employment

Pra1: Categorical variable which represents the working situation of the individual. The identification level assigned to this variable is 0 and it is specified as follows:

0, NSP (Non-Sampling Population)
1, Employed
2, Unemployed having worked
3, Unemployed seeking 1st job
4, Inactive

Tjor: Categorical variable which represents the type of workday of the employee. The identification level assigned to this variable is 0 and it is specified as follows:

0, NSP(Non-Sampling Population)+Inactives+Unemployed
1, Full-Time
2, Part-Time

Htrt: Numerical variable which represents the total weekly working hours of the individual.

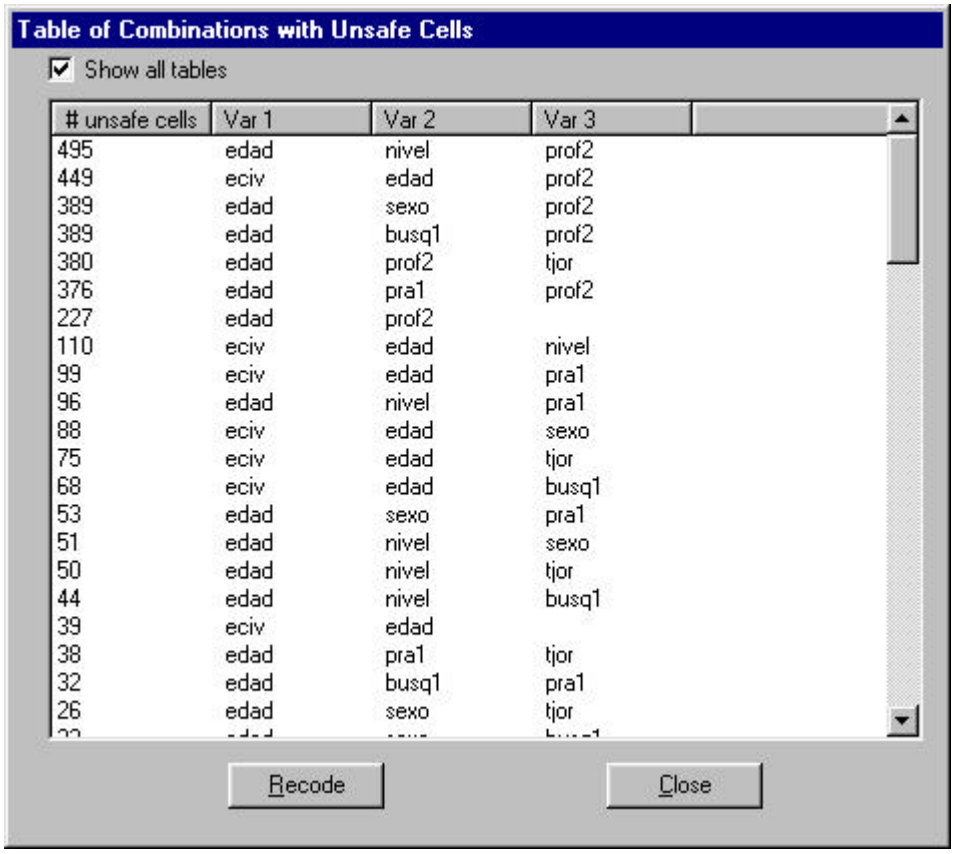
Dpar1: Numerical variable which represents the length of unemployment (months) for each individual.

Eleva: Weight variable which represents the weighting quantities corresponding to each individual.

We shall study the identification keys of dimension 3. That is, μ-Argus generates all possible frequency tables by crossing 3 identifying variables (in this example, all the categorical ones) and it counts the number of unsafe cells for each variable combination.

In this example a cell is considered to be unsafe if it is a unit cell (our threshold is D_k=1), that is, we shall determine the number of individuals who are unique with respect to a certain key. Being aware of the keys which generate the highest number of unit cells, we could make decisions about variables which need to be recoded or transformed in order to reduce the number of unsafe cells.

This is the informative screen that μ-Argus shows once all possible combinations of 3 variables have been generated:



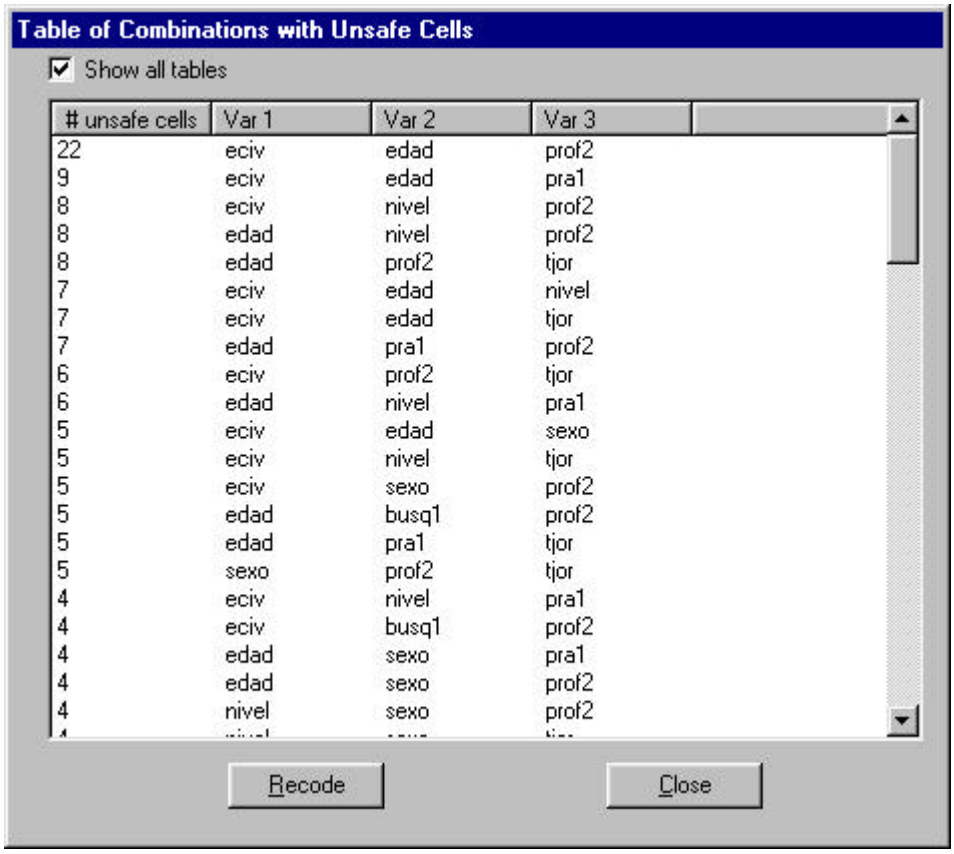
# unsafe cells	Var 1	Var 2	Var 3
495	edad	nivel	prof2
449	eciv	edad	prof2
389	edad	sexo	prof2
389	edad	busq1	prof2
380	edad	prof2	tjor
376	edad	pra1	prof2
227	edad	prof2	
110	eciv	edad	nivel
99	eciv	edad	pra1
96	edad	nivel	pra1
88	eciv	edad	sexo
75	eciv	edad	tjor
68	eciv	edad	busq1
53	edad	sexo	pra1
51	edad	nivel	sexo
50	edad	nivel	tjor
44	edad	nivel	busq1
39	eciv	edad	
38	edad	pra1	tjor
32	edad	busq1	pra1
26	edad	sexo	tjor

We can see that the unsafest key is *edad x nivel x prof2* since it is the combination which generates a higher number of unsafe cells. Additionally, the variables which appear more often in the unsafest combinations are *edad* and *prof2*. Assisted by the program, we shall make a *global recoding* of both variables to reduce the number of unsafe cells. New recodings for these variables are the following:

Edad:
1, < 16 years old
2, [16-24]
3, [25-34]
4, [35-44]
5, [45-54]
6, [55-64]
7, > 64 years old

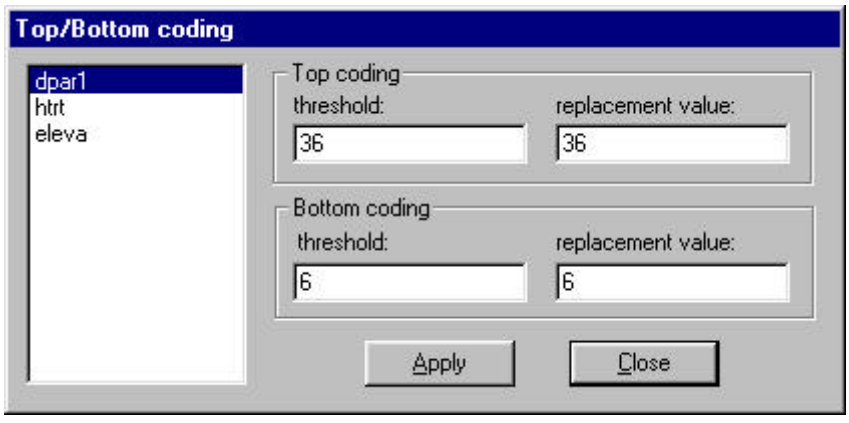
Prof2:
0, NSP
1, Professionals and Technicians
2, Managerial staff
3, Administrative staff
4, Merchants and salespersons
5, Service staff
6, Farmers
7, Industrial workers

After global recodings, we can see that the number of unsafe cells has decreased noticeably.



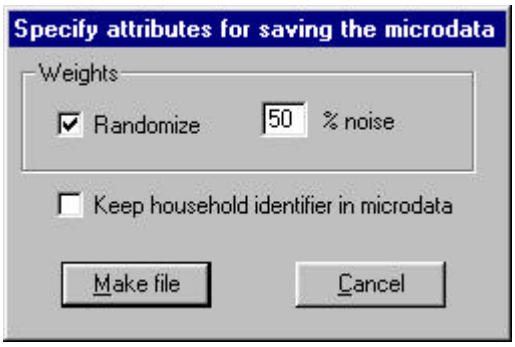
The unsafe cells left will be protected by *local suppressions*. This procedure is carried out by μ-Argus, automatically and optimally when we create the safe file at the end of the process.

Before finishing, we can apply any of the available protection techniques for numerical variables. In our case, we apply the *top-bottom coding* to variable *dpar1:*

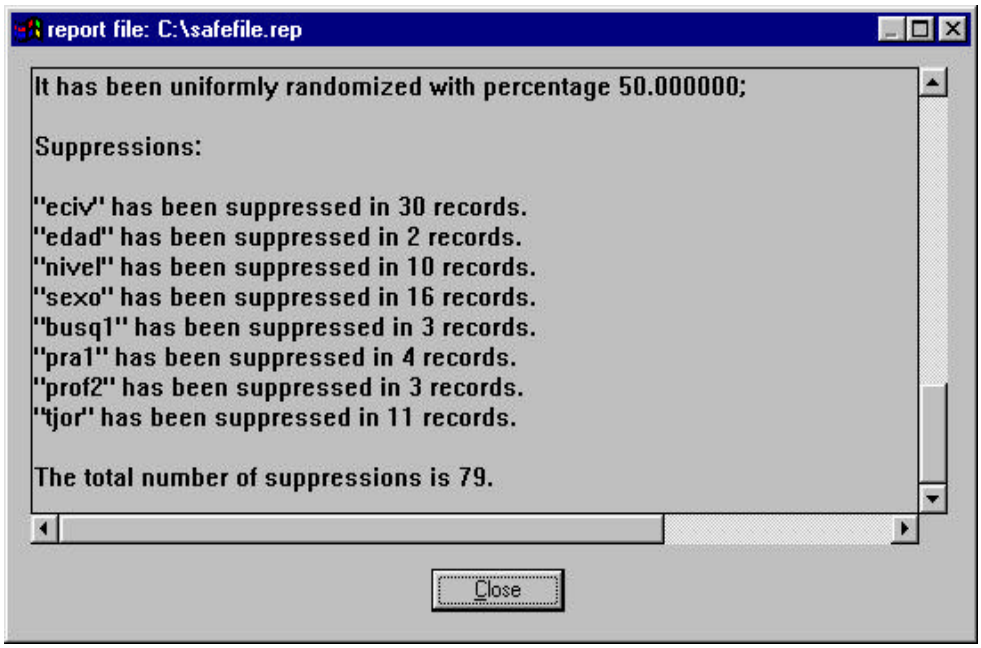


The result of the top coding is that, for all the records having value of its *dpar* variable more or equal to 36, this value will be replaced by the replacement value (in our case 36). On the other hand and as a result of the bottom coding, for all the records having value of its *dpar* variable less or equal to 6, this value will be replaced by the replacement value (in our case 6). This type of coding tries to hide extreme values in data which are more sensitive than others. For known distributions (Normal, Chi-square,...) these replacement values correspond to those which limit the distribution tails (e.g. $\pm 2\sigma$ in Normal case).

Finally, we only have to create the protected file with all the recodings and transformations carried out on the variables. At this stage, μ-Argus asks for the weight variable to be randomised in a certain percentage before publication. In addition, we could decide to publish or not the household identifier. For the example, the weight variable is *Eleva* and we are going to randomise it in 50% noise:



When the safe file is created, Argus makes the local suppressions needed to avoid the identification of unit cells, always minimising the information loss. That is, if there is more than one unsafe combination in a record and the unsafe combinations have a variable in common, then μ-Argus will suppress the common variable. If not, μ-Argus will have to choose one of the variables minimising the information loss. For our example these are the suppressions carried out automatically by Argus:



From the original file with 32,043 total data entries (11 variables x 2913 records), we have obtained a safe file with respect to three-dimensional keys in which we have only suppressed 79 data entries.

It is possible to protect datasets against keys of higher dimensions or against specific combinations with special high risks of disclosure. Choosing which key is to be analysed should be the responsibility of the data protector who is assumed to have certain prior knowledge of the population and knows better what could suppose a higher or lower risk of disclosure.

Chapter

4

Tabular Data Protection

Granularity

The concept of *granularity* is explained in [14] drawing a comparison between the observation of tabular data and the process of producing a photographic image. If we amplify indefinitely a photograph we will be able to see only points or "grains", losing definition and the global image. The observer's attention is diverted to the photographic production process, he stares at the detail. The same thing could happen in a table of data where the ultimate aim is to inform about more or less aggregated characteristics of the population (global image) instead of giving detailed attributes of individuals (amplifying image). If the table contains a lot of unit cells (only one record contributes to the cell value) the attention is diverted to individual responses with the consequent high risk of identification. The level of "granularity" of a table will help us to recognise if it is safe or not and to determine if it is necessary to apply protection measures to avoid identity disclosures.

Definition and Quantification

Summarised data release in tabular format does not supply exact information concerning each individual in the population but does tell us about the contributors to each cell and their distribution for a certain row or column in the table.

In contrast to microdata files, where the disclosure of a unique record endanger data security, the risk of disclosure in tabular data will be given by the unit cell density within a certain *hyper-row.*(*)

Granularity can be defined as *a safety criterion for tabular data, based on the number of unit cells or "grains" contained in it, and their distribution with respect to other cells in the same hyper-row.*

We consider a table as being safe if it does nott reach certain *index of granularity* defined for each hyper-row as follows:

$$Index\ of\ Granularity = \frac{number\ of\ unit\ cells}{total\ number\ of\ cells\ with\ value} \times 100$$

If this index is under 50%, no action on the hyper-row is required, otherwise it is. A decision process, which decides if the table is safe or not, has to take into account the granularity incidence over all the possible hyper-rows generated by fixing categories of identifying variables.

(*) Subset of cells originating from crossing identifying variables and defined by fixing categories in one or more of these variables. E.g. Occupation x Region x Sex define an hyper-row for the categories Sex=female and Occupation=Statistician.

Qualitative Attributes of Granularity

Apart from the quantitative aspect of granularity, some qualitative characteristics influence the granularity, affecting to the risk of disclosure:

- *Degree of clustering of observations.* The higher the degree of clustering, the more dispersed the population distribution is along a hyper-row, and the higher the risk of finding unit cells.

- *Unexpected Patterns.* A hyper-row can be seen as an array of cell frequencies which defines the sample distribution of a subpopulation. If these frequencies are unusual with respect to the population distribution (that is, they do not fit the population pattern), this could mean a high index of granularity.

Types of Granularity

To define different types of granularity we will use an illustration. Let *statistician x age x female* be the hyper-row of a hypothetical tridimensional table defined by Occupation x Age x Sex. Let us consider that it has been tabulated in a population of a medium sized provincial town placed near a large administrative centre.

- The following hyper-row would represent a normal situation, in which the granularity index and, consequently, the disclosure risk are low.

Normal.- Ungranulated

Sex=female	AGE				
	5-19	20-34	35-50	50-65	66+
...					
Occupation=statistician	1	30	18	9	2
...					

The unit cell in the 5-19 age interval does not add any risk of disclosure since the hyper-row distribution fits the expected population distribution perfectly.

- *Proportional Granularity*. Is that which is greater than or equal to a certain critical level or index of granularity (in our case determined by 0.5).

Proportional.- Index ³ 0.5

Sex=female	AGE				
	5-19	20-34	35-50	50-65	66+
...					
Occupation=statistician	1	3	4	1	1
...					

- *"Odd" Granularity or unusual pattern*. There are unexpected unit cells which are "rare" with respect to the characteristics of the population or which do not fit the "a priori" population distribution.

"Odd".- Unusual Pattern

Sex=female	AGE				
	5-19	20-34	35-50	50-65	66+
...					
Occupation=statistician	0	1	1	21	1

In this case, a higher frequency in the 20-34 age interval would be expected and the same for the 35-50 age interval.

- *Dispersed Granularity*. This contains isolated unit cells between high frequency cells.

Dispersed.

Sex=female	AGE				
	5-19	20-34	35-50	50-65	66+
...					
Occupation=statistician	1	30	1	9	1
...					

Depending upon the different types of granularity, the corresponding critical granularity patterns can be created. If a table presents any of these patterns, this will indicate a high level of granularity and then we could apply the specific measures against the risk of disclosure of individual information.

Application to Electronic Micro-Tables

An Electronic Micro Table (or EMT) refers to the *public release of a highly disaggregated multi-dimensional table on electronic medium*. They are used to model richer population structure and they are very useful to explore explanatory relationships in the population, where publication by means of microfiles could suppose a high disclosure risk.

Risk of identification in EMT is given by the unit cell distribution. At this point, the concept of granularity as unsafe table detector makes sense.

A EMT can be seen as a record file where fields are each table variable and records are defined by the values of each cell in a hyper-row. This equivalence makes the microfile protection techniques *(see Chapter 2)* useful to detect granularity in micro tables.

This is the case of μ-Argus, which is based on the detection of "rare" individuals (unique population) and could be adapted to detect unit cells within micro tables. The following algorithm is an implementation prototype which combines granularity detection in tables with a specific disclosure control procedure for microfiles such as μ-Argus:

Algorithm to eliminate granularity in microdata files

Input. Microdata file with confidential records.

Otput. Microdata file protected by granularity criterions.

Step 1. μ-Argus generates two-dimensional and three-dimensional tables for the n identifying variables (key variables) considered.

Step 2. If cell unit proportion in any given hyper-row, which is constructed by fixing values of the identifying variables, is ≥ 0.5 then the whole table is considered unsafe.[*]

Step 3. If the table is considered unsafe, affected rows and columns are redefined by aggregating categories or suppressing any unit cell.

Step 4. Step 4 is repeated until granularity falls below a determined level of security for each hyperfile and unsafe table.

Step 5. The file of protected date is reconstructed according to granularity criteria and produced via μ-Argus.

The application of granularity criteria for safe data file creation, for those cases for which the optimum solution has not yet been found prevents the occasional excessive loss of information which is often caused by cell suppression methods and gives an approximation of an optimum solution.

On the other hand if a file generated in this way is safe, so will any derived table, thus we have a file creation method which may be used externally (in external databases), in which the user constructs the tables which are of use according to his or her needs.

Rounding Method

This method is simple and easy to use, in both frequency and magnitude tables of not too many dimensions (double or triple entry). This technique enables the choice of an entire data base, on which the rounding off is carried out, always in a controlled manner, as the rounding of tables may effect the additivity of the totals or subtotals.

Definition 1. *Given a integer rounded data base b, the rounding process of Table A consists in substituting each table frequency or total for one of the two integer multiples of b which are closest to the cell value. If this value is already a multiple of b, the cell itself and the following multiple higher than b will be taken as adjacent values.*

In the case of bidimensional tables, rounding off may be restricted to non-nullity conditions, this supposes that if the entry is already a multiple of b, its fixed value is maintained.

The problem when passing to three or more dimensions is not easy to solve. In theory, the tables may be of any number of dimensions although in practice two or three dimension tables are more common. In Ernst(1989)[*] a solution is proposed for three dimensional tables based on *successive rounding*, that is rounding off the rounding value. This is an efficient although not widely used method which also respects the nullity conditions mentioned above.

Cell Suppression Systems

Cell Suppression Systems are one of the protection mechanisms for tables most widely used by statistics offices and agencies. They are generally applied in *aggregated*

[*] We could apply any other index of granularity either qualitative or quantitative if we know the type of granularity which affects the population.

[*] Ernst, L.R.(1989) "Further applications of linear programming to sampling problems". Technical Report-Census SRD (RR-89-05)

magnitude tables although they may also be adapted to the case of *frequency tables* . These techniques are not so easily automated as rounding techniques as not only is suppression of confidential or sensitive cells necessary, but it is also necessary to carry out a *secondary suppression* of the cells which may provide information concerning those eliminated by the first pattern of suppressions.

Currently the primary suppression of cells is carried out automatically, however the second part of the process is carried out manually in many cases. In this section we shall attempt to explain the keys towards the automation of an efficient system for the secondary suppression of cells and we will deal with the explication of a model of integer linear programming provided by the optimum pattern of suppressions in terms of information loss.

Sensitivity Measures

Before dealing with a suppression of cells, it is necessary to determine which of these imply a disclosure risk according to certain *sensitivity rules.* The application of one or another rule will affect the pattern of both primary and secondary. These are a few of the rules:

- **Rule (n, k).** A cell will be suppressed if the n contributions greater than the value of the cell constitute more than k% of its total value.
- **Percentage p Rule.** A cell will be suppressed if a user is capable of approximating the contribution of a certain individual to the value of the cell to within ± p%
- **Rule p-q.** This is an extension of the above, in which any previous knowledge of the population which the user may have is taken into account, measured again in the contributions of an individual to a cell.

All the above rules are based on the contribution of individuals to the value of the cells. If a cell is "dominated" by one or two individuals, then more protection is required.

Measuring Information Loss

Obviously, when applying a suppression pattern the objective is to minimise information loss. To this end a loss function is specified which may be given by:

- **The number of cells suppressed.** The optimisation of this function will lead to the suppression of the least number of high value cells.
- **The sum of the values of the suppressed cells.** To minimise this function it is preferable to eliminate a greater number of lower value cells.

Depending upon the purpose of the table, one or other loss function must be chosen, obtaining a different suppression pattern for each case. Often this pattern has to be adjusted to the needs of the user or client who requires the information, thus the system has to permit "marking" of certain cells which may be of greater use or importance to the user, in such a way that there is a minimum probability that these are included in the suppression pattern.

A Mixed Integer Linear Programming Model for Secondary Cell Suppression **[8]**

As a way of presenting the problem we shall look at a simple example for a two dimensional used by Willenborg and Waal:(*)

Example

The following tables show the investments made by companies (in millions of monetary units) in each region in different activities during a certain period:

	A	B	C	Total
Activity I	2	50	10	80
Activity II	8	19	22	49
Activity III	1	32	12	61
Total	4	10	44	190

(a) Original table

	A	B	C	Total
Activity I	20	50	10	80
Activity II	*	19	*	49
Activity III	*	32	*	61
Total	45	10	44	190

(b) Published table

Assuming that the information corresponding to Activity II in region C is confidential, then the cell with the value of 22 is considered sensitive and must not be published. However this is not enough, as due to the presence of totals and subtotals it is possible to calculate its value. Therefore it is necessary to withhold other values in the table (see b). In this way the exact value of the confidential cell cannot be calculated, even though it is possible to determine a range of possible values for the cell which are consistent with those published.

If we let $y_{23}(\inf)$ be the minimum possible value of the sensitive cell, this may be calculated by solving the following linear programming problem, in which the unknown y_{ij}' represent the suppressed values (i,j) in the table:

$$
\begin{aligned}
& y_{23}(\inf) = \min \quad y_{23}' \\
& s.a. \\
& \quad y_{21}' + y_{23}' = 30 \\
& \quad y_{31}' + y_{33}' = 29 \\
& \quad y_{21}' + y_{31}' = 25 \\
& \quad y_{23}' + y_{33}' = 34 \\
& y_{21}', y_{23}', y_{31}', y_{33}' \geq 0
\end{aligned}
$$

Similarly the maximum value $y_{23}(\sup)$ can be calculated by changing the target function and using the same restrictions.

In this case the solutions are $y_{23}(\inf)=5$, $y_{23}(\sup)=30$ and we can say that the sensitive information is protected within a range of [5,30]. If this interval is considered to be sufficiently wide by the statistics agency or institute, then the cell is considered to be protected. Sometimes so-called *external bounds* may be applied in order to stretch this interval. These are calculated supposing that the variation in the nominal value of the sensitive cell will be no greater than a certain percentage. For example, if in our case we

(*) Willenborg and Waal, "Statistical Disclosure Control in Practice", Lecture Notes in Statistics 111, Springer, New York, 1996.

limit the variation in the sensitive value to ±50%, that is we add the condition $11 \leq y_{23}' \leq 33$ to the linear programming problem, we obtain a more realistic protection interval [18.26].

Note:

Given a group of cells SP *(primary suppressions)* together with each ones required protection levels (fixed by the statistics agency or institute), the final objective consists in calculating the optimum set of secondary suppressions which protect all the sensitive cells from a possible attack, and in a way that the information loss resulting from these suppressions is minimal..

Let:

$i = \{1,.....,n\} \rightarrow$ set of table indexes to be protected.

$w_i \geq 0$ the cost of the suppression of cell i

$[a_i] \rightarrow$ values for the original table entries *(nominal table)*

$[y_i] \rightarrow$ possible values for the table entries *(possible table)*

Let us say that a vector $[y_i]$ defines a possible table when $Ay=b$, where A is a matrix {0,1,-1} and b=0

Let us assume that the attacker knows a range of possible values (external bounds) for each entry for the table a_i , that is $[a_i-lb_i, a_i+ub_i]$ and $a_i-lb_i \leq y_i \leq a_i+ub_i$ for all i. .

Let $i_1,....i_p$ with $p = |SP|$, the indexes of the set of sensitive cells SP.

If $[a_{ik} - LPL_k,\ a_{ik} + UPL_k] \subset [y_i(inf),\ y_i(sup)]\ \ \forall\ k=\{1,.....,p\} \Rightarrow$ safe table

Where LPL_k y UPL_k are lower or greater than the levels of protection required by the statistics agency or institute and y $[y_i(inf),\ y_i(sup)]$ is the interval calculated by a possible attacker.

The Model

Given a set SUP of suppressions (primary and secondary), the problem of a possible attacker consists in calculating maximum and minimum values for the cells included in SP within a table in which only the published values coincide with the originals from the nominal table. For each i_k we will have a subproblem of the following type :

$$y_{ik}(inf) = \min y_{ik}'$$

$$Ay' = b \qquad (1)$$

$$a_i - lb_i \leq y_i' \leq a_i + ub_i \quad \text{para } i = 1,.....,n$$

$$y_i' = a_i \quad \text{para todo } i \notin SUP$$

$y_{ik}(sup) = \max y_{ik}''$

$Ay'' = b$ (2)

$a_i - lb_i \le y_i'' \le a_i + ub_i$ for $i = 1,.....,n$

$y_i'' = a_i$ for all $i \notin SUP$

Where the continual variables y' and y'' are local for each subproblem.

To be able to formulate the above as a mixed integer linear programming model we introduce a binary variable X defined as follows:

$x_i = 1$ if $i \in$ SUP (suppressed cell)

$x_i = 0$ otherwise

Thus we obtain the following function to be minimised::

$$min \sum_{i=1}^{n} w_i x_i$$

Subject to:

$x \in \{0,1\}^n$ and for each $i_k \in SP$, $x_{ik} = 1$

$LPL_k - a_{ik} \le -y_{ik}(inf) = \max(-y_{ik}')$

$Ay' = b$ (3)

$a_i - lb_i x_i \le y_i' \le a_i + ub_i x_i$ for $i = 1,.....,n$

$LPL_k - a_{ik} \le -y_{ik}(sup) = \max y_{ik}''$

$Ay'' = b$ (4)

$a_i - lb_i x_i \le y_i'' \le a_i + ub_i x_i$ for $i = 1,.....,n$

In order to solve such a complex problem, in which there are a great number of local variables (y' and y'') and interrelationships between these and the introduced variable X, the "relaxation" of the model is necessary, (that is, permit the condition $x_i \in \{0,1\}$ to be converted in $(0 \le x_i \le 1)$. Furthermore we will set out a linear programming model, which we shall call *problem master*, which depends initially only on the variables x_i:

$$min \sum_{i=1}^{n} w_i x_i : x_{i_1} = x_{i_2} = ... = x_{i_p} = 1,\ x \in [0,1]^n$$

Let x^* be the optimum solution found for the previous problem. This is substituted in (3) and (4) in what we call the *verification procedure*, which consists in solving of the 2p small problems of linear programming, which will return local vectors y' and y" as results fulfilling the conditions imposed by the protection limits and confirming the optimality of x^*. If x^* is not possible, the procedure will have found, for some of the problems, a value strictly below than the protection limits imposed $LPL_k - a_{i\,k}$ or $LPL_k + a_{i\,k}$, and for each dual solution of these problems a linear inequality may be derived (only as a function of x) which is not fulfilled by x^*, thereby proving that x^* is not a possible solution of the complete problem.

These new inequalities, called *Benders bounds*, are added to the master problem and this is optimised again (through parametric techniques e.g. simplex algorithms). The process is repeated until an optimum solution x^* is found for the relaxed model. If this solution is complete, it will be the solution of the initial problem of integer linear programming. Otherwise, branch-bounding methods have to be applied in order to force the integrity of x^*.(*)

The Benders bounds calculated during this process are in this form:

$$\sum_{i=1}^{n} (b_i' ub_i - a_i' lb_i) x_i \geq LPL_k$$

$$\sum_{i=1}^{n} (b_i'' ub_i - a_i'' lb_i) x_i \geq LPL_k$$

Where (α',β') and (α",β") are dual optimum solutions for the problems (3) and (4).

These bounds, also called *capacity bounds*, may be strengthened by replacing $(b'_i\, ub_i - a'_i\, ub_i)$ with $min\{(b'_i\, ub_i - a'_i\, ub_i), LPL_k\}$ and $(b''_i\, ub_i - a''_i\, ub_i)$ for $min\{(b''_i ub_i - a''_i\, ub_i), LPL_k\}$, which turns out to be very effective in practice when it comes to reducing the number of process repetitions.

Other important Keys

Other factors consider when it comes to implementing an efficient system of cell suppression are as follows:

- **Treating "zeros".** A cell with the value 0, is providing precise information about the characteristics which certain individuals **do not** have, which also implies a disclosure risk if this data is confidential. Protecting a cell with zero contributions is not a trivial problem. According to any of the sensitivity rules stated above, a zero cell would never be considered sensitive (obviously there is no dominant contribution to the value of the cell), thereby requiring a special treatment which the suppression system has to take into account.

(*) M.Padberg, G,Rinaldi, "A branch-and-cut algorithm for the resolution of large scale symmetric travelling salesman problems", SIAM Reviews,33 (1991)

- **Simultaneous Suppression.** A system of simultaneous suppression searches for the best pattern of suppressions which protect all the sensitive cells "once only". However it is only applicable in the case of very simple tables, resulting quite inefficient in terms of time as it checks all the possible suppression patterns for each table.

- **Sequential Suppression.** This protects primary suppressions, sequentially, assuring protection at each step of the process. This system tends to overprotect if it does not have a "memory" which collects the cells included in the suppression pattern. In this way the system may use previously suppressed cells at each step in such a way that redundant or unnecessary suppressions are avoided.

- **Multiple Dimensions.** Linear Programming methods have provided an efficient and valid solution for protection in multi-dimension tables. Even when these do not provide the optimum solution, they do give an approximation and in many cases it can be "refined" through the application of heuristic methods.

- **Multiple Tables.** The one way of guaranteeing efficient protection between tables consists of producing all of them from a unique data set. The suppression system must be able to detect common cells in multiple tables which may often not be produced simultaneously. This assumes maintaining a memory of all the suppressions carried out on previous tables, with the subsequent requirements for control and storing capacity.

τ- Argus for table protection

τ-Argus is the specific module for the protection of integrated tables in the software package ARGUS [23]. It enables us to create "safe" tables from data file plans by applying the most commonly used techniques in the tabular data protection.

The principal methods used by τ-Argus in the process will be *controlled rounding* and *cell suppression* (both primary and secondary). For both of these it will use the integer linear programming model described in the previous section. It will also permit global re-coding of the variables before applying the aforementioned techniques, in such a way that the number of sensitive or confidential cells generated by the application of sensitivity criteria is reduced as much as possible (in this case τ-Argus applies the sensitivity rule (n,k)).

How does τ-Argus work?

Working phases of τ-Argus:

1. Reading the data files and description of variable files.

2. Specification of the table to be protected (up to 3 dimensions).

3. Indication of the sensitivity criteria.

4. Creating the table and determining the number of sensitive or confidential cells.

5. Application of the global re-coding method until the number of sensitive cell is minimised.

6. Application of secondary cell suppression, rounding or both of these, in order to protect the sensitive cells.

7. Creation and storing of the safe table.

Example

An example using data from an economic survey carried out by EUSTAT appears below. The population is made up of 313 companies which carry out R&D Euskadi.

1. The data corresponds to 1996 and the following variables are used in the process:

act18: Category variable representing the activity carried out by the company classified in 18 activity branches.

tamaño: Category variable representing the size of the company according to the number of employees divided into 7 size categories.

gastot: Numeric variable representing the total financial costs of the company in millions of pesetas.

2. We shall specify the bidimensional *act18 x tamaño*. We shall set up the frequency table (number of companies which contribute to each cell) and then a magnitude table for the same category variable cross, representing the variable *gastot* as an item within each cell.

3. The following will serve as sensitivity criteria:

- For the frequency table those cells with a value ≤ 2 will be considered as sensitive cells. (two or fewer companies contribute to the cell)

- For the magnitude table those cells in which 2 or fewer companies contribute to the value will be considered as sensitive along with all those whose dominant contribution accounts for more than 75% of the total value of the cell.

4. Below we shall see the tables created by Argus and which cells are considered to be sensitive according to the specified criteria:

tamaño act18	<25	25-49	50-99	100-249	250-499	500-999	>=1000	total
Agriculture and fishing	**2**	.	.	.	.	.	.	**2**
Chemical	7	3	6	**2**	5	**1**	.	24
Rubber and Plastics	**2**	.	3	8	3	.	**1**	17
Metallurgy	.	**2**	**1**	7	6	**2**	**2**	20
Metal Products	5	.	7	6	8	4	**1**	31
Machine tool	.	4	9	4	**2**	**1**	.	20
Domestic Apparatus	.	.	**2**	**2**	**2**	**1**	**1**	8
Other machinery	**1**	4	13	11	3	**1**	.	33
Electric equipment	5	6	**2**	3	5	.	.	21
Electronic equipment	4	**2**	**2**	**2**	**2**	.	.	12

Precision equipment	12	**2**	3	**2**	**1**	.	.	20
Transport material	.	**2**	**2**	4	6	.	**2**	16
Other manufacturing	3	4	7	5	**2**	.	.	21
Energy and construction	**1**	**1**	**2**	**3**	**1**	.	**2**	10
IT activities	5	**2**	3	**1**	.	.	.	11
R&D Activities	6	3	3	6	.	.	.	18
Other business activities.	14	5	3	**1**	**2**	**1**	.	26
Other services	**2**	.	.	**1**	.	.	.	3
Total	69	40	68	68	48	11	9	313

The cells in bold type are sensitive or confidential from the frequency tables. As we can see the example chosen is very sensitive as the population is not particularly large (313 companies) and is considerably disintegrated. We have to re-code both variables in order to reduce the number of sensitive cells.

For the magnitude table (the items within the cell correspond to the numeric variable *gastot*) we will have the same number of sensitive cells + those which fulfil the (n,k) criteria which is imposed.

5. The variables *act18* and *tamaño* will be re-coded in the following way:

act18 will change to have 9 categories

tamaño will be re-coded in 2 categories (companies with <100 employees and the rest)

The magnitude table created by Argus with these new re-codings appears below (the sensitive cells are in bold type)

Total financial costs (gastot)

Tamaño act18	<= 100 workers	>100 workers	Total
Agriculture and fishing	**22196**	.	**22196**
Chemical	593301	1425519	2018820
Rubber and Plastics	388895	486771	875666
Metal	834529	2962455	3796984
Machinery	2274677	2740140	5014817
Materials and manufacturing	4032117	8563599	12595716
Energy and construction	207072	350013	557085
Other activities	16535820	**1726625**	18262444
Other services	**171819**	.	**171819**
Total	25060426	18255122	43315548

The number of sensitive cells are been reduced considerably. The cells in bold type will not be published and a complementary suppression will protect those which are to be recalculated.

6. Argus provides us with a pattern of secondary suppressions which protects confidential data within the security limits laid down by the statistics agency or institute. In

our case 70% and 130% (that is to say we protect the cells within 305 of their nominal value).

The pattern of suppressions is also calculated according to a variable cost of information loss. In our case the variable cost will be the same as that represented in the cells (*gastot*) this assumes that the greater the value of the variable in the cell the lower the probability that this will be suppressed and therefore the lower the information loss. The following table reflects the optimum suppression pattern according to Argus:

Total financial expenses (gastot)

Tamaño act18	<= 100 workers	>100 workers	Total
Agriculture and fishing	**s**	.	**s**
Chemical	593301	1425519	2018820
Rubber and Plastics	388895	486771	875666
Metal	834529	2962455	3796984
Machinery	2274677	2740140	5014817
Materials and manufacturing	4032117	8563599	12595716
Energy and construction	**s**	**s**	557085
Other activities	**s**	**s**	18262444
Other services	**s**	.	**s**
Total	25060426	18255122	43315548

7. It can be seen that Argus has added 3 complementary suppressions to those already existing. This table will be published and considered safe according to the imposed limits of security.

There are multiple ways of securing a table, as many as there are possible ways of re-coding or different specifications we wish to use (sensitivity criteria, security limits....). The flexibility of Argus will enable us to create safe tables both for standard and made to measure publications for a particular user, without the data losing statistical usefulness or information capacity.

Chapter

5

Conclusions and the Future

The intention of drawing up this document was to review the latest techniques developed in the field of Security and Protection of statistical data.(*) Furthermore, the focus of attention has been towards the strictly statistical aspect of confidentiality, identifying the existence and the continuing development of methods based on the data itself, which permit maximum quality of information in the most secure manner.

Technique Development

Advances in this field are occurring at a great velocity and at the time of writing this document, new techniques are being developed offering an improvement and an alternative to the methods outlined here. Following is a list of certain common aspects which have to be taken into account when applying these new techniques, both regarding their application in register files and in tables and databases:

- To embrace all the production phases of statistical data (collection, analysis and disclosure)

- To avoid over-protection, that is, do without unnecessary or redundant suppressions in tables, or severe re-coding in data files which lead to excessive information loss.

- To study probabilistic models which provide quantitative measures of disclosure risk, thereby enabling comparisons between safe files or tables and aiding the choice of the best option depending upon each case and needs of the user.

Software and Information Technology Procedures

It is impossible to ignore the parallel development of a standardised software which encompasses the implementation of all these techniques and which, in turn, is compatible with new information systems and international networks of communication.

New versions of the ARGUS software presented here are being developed as part of the a new European project concerning the control of statistical data disclosure. One of the improvements which is being studied is the application of micro-aggregation for several variables in the μ-Argus model, and working with groups of related tables in the τ-Argus module. The compatibility of entries and output for related databases (Access, Oracle,…) is also being studied.

Not only at European level but also throughout the world there are specialised software packages which include the latest advances with regard to techniques and which provide ever increasing effectiveness in execution times [11].

(*) The majority of the methods outlined were presented at the Security and Protection of Statistical Data Congress held in Lisbon on March, 1998.

Bibliography

[1] Baeyens,Y., Defays, D. (1998) "Estimation of variance loss following microaggregation by the individual ranking method". Proceedings of Statistical Data Protection 98' Lisbon.

[2] Cox,L.H., Zayatz, L..V. (1995) "An Agenda for Research in Statistical Disclosure Limitation". Journal of Official Statistics, Vol. 11, No.2, pp.205-220.

[3] Cox,L.H. (1998) "Some Remarks on Research Directions in Statistical Data Protection". Proceedings of Statistical Data Protection 98' Lisbon.

[4] Domingo-Ferrer,J. Mateo-Sanz,J.M. (1998) "A method for data-oriented multivariate microaggregation". Proceedings of Statistical Data Protection 98' Lisbon.

[5] Domingo-Ferrer,J., Mateo-Sanz,J.M., Sánchez del Castillo, R.X. (1999) "Cryptographic Techniques in Statistical Data Protection". Joint ECE/Eurostat Work Session on Statistical Data Confidentiality. Thessaloniki, Greece 99'.

[6] Domingo-Ferrer,J., Sánchez del Castillo, R.X., Castilla,J. (1998) "Dike: A Prototype for Secure Delegation of Statistical Data". Proceedings of Statistical Data Protection 98' Lisbon.

[7] Fienberg, S.E. (1994) "Conflicts Between the Needs for Access to Statistical Information and Demands for Confidentiality". Journal of Official Statistics, Vol. 10, No.2.

[8] Fischetti,M., Salazar,J.J. (1998) "Modelling and Solving the Cell Suppression Problem for Linearly-Constrained Tabular Data". Proceedings of Statistical Data Protection 98' Lisbon.

[9] Franconi, L.. (1999) "Level of safety in microdata: Comparisons between different definitions of Disclosure Risk and Estimation Models". Joint ECE/Eurostat Work Session on Statistical Data Confidentiality. Thessaloniki, Greece 99'.

[10] Garín, A., Ripoll, E. (1999) "Performance of μ-Argus in Disclosure Control of Uniqueness in Populations". Joint ECE/Eurostat Work Session on Statistical Data Confidentiality. Thessaloniki, Greece 99'.

[11] Giessing, S. (1998) "Looking for Efficient Automated Secondary Cell Suppression Systems: A Software Comparison". Proceedings of Statistical Data Protection 98' Lisbon.

[12] Gopal,R., Goes,P. (1998) "Confidentiality via Camouflage:The CVC approach to Database Query Management". Proceedings of Statistical Data Protection 98' Lisbon.

[13] Holvast, J. (1999) "Statistical Confidentiality at the European Level". Joint ECE/Eurostat Work Session on Statistical Data Confidentiality. Thessaloniki, Greece 99'.

[14] Horn,S., Morton, R., (1998) "Protecting Output Databases". Proceedings of Statistical Data Protection 98' Lisbon.

[15] Hundepool,A . J., Willenborg,L. (1998) "ARGUS for Statistical Disclosure Control". Proceedings of Statistical Data Protection 98' Lisbon.

[16] Hundepool,A . J., Willenborg,L. (1999) "ARGUS: Software from de Statistical Disclosure Control Project". Joint ECE/Eurostat Work Session on Statistical Data Confidentiality. Thessaloniki, Greece 99'.

[17] Jabine, T.B. (1993) "Statistical Disclosure Limitation Practices". Journal of Official Statistics, Vol. 9, No.2, pp.436-454.

[18] Keller-McNulty,S., Unger, E.A . (1993) "Database Systems:Inferential Security". Journal of Official Statistics, Vol. 9, No.2, pp.475-499

[19] Lambert, D. (1993) "Measures of Disclosure Risk and Harm". Journal of Official Statistics, Vol. 9, No.2, pp.313-331.

[20] Malvestuto, F.M., Moscarini, M. (1998) "An Audit Expert for Large Statistical Databases". Proceedings of Statistical Data Protection 98' Lisbon.

[21] McLeod, K.., George.J., Rae, A., Butler,R. (1998) "Investigating Key Qualities of an Automated Cell Suppression System". Proceedings of Statistical Data Protection 98' Lisbon.

[22] μ-ARGUS.Version 3.0. User´s Manual. Contributors: Hundepool, A.J., Willenborg,L.., Wessels, A., van Gemerden, L.., Tiourine, S., Hurkens,C.

[23] τ-ARGUS. Version 2.0. User´s Manual. Contributors: Hundepool, A.J., Willenborg,L.., Wessels, A., van Gemerden, L.., Fischetti, M., Salazar,J.J., Caprara, A.

[24] Ley de la Función Estadística Pública (9 de Mayo de 1989) Tit.1 Cap.III "Del secreto estadístico".