Journal of Official Statistics
Vol. 2. No. 3, 1986, pp. 329–336

Finding a Needle In a Haystack

or

Identifying Anonymous Census Records

Tore Dalenius[1]

1. Introduction

1.1. Posing the problem

A population census has just been brought to completion. The collected and processed data are recorded on a processing medium like punchcards or magnetic tape. In the discussion that follows, we assume that there is one data record for each individual enumerated in the census.

We wish to preserve the census data since they may play an instrumental role in the design of future sample surveys. They could, for instance, be used in the creation of a sampling frame, or for some additional analysis. A condition for preserving the data is that it not be possible to link a record with the corresponding individual, thus making the invasion of this individual's privacy possible. It is well known that making records anonymous by removing formal identifiers such as name, address, and population registration number, 'de-identification', while necessary, may not be sufficient to eliminate the possibility of such a linkage. This state of affairs may be explained in the following technical terms. The data for some variables may, for some individuals, be unique and publicly known; hence they may serve as 'quasi-identifiers', i.e., make it possible to link the records with the corresponding individuals.

[1] Brown University, Providence, R.I., U.S.A.

1.2. The purpose and organization of this paper

This paper has a genuinely modest purpose, viz. to present two variants of a simple method for identifying unique records. Both variants have in common that they entail sorting the records. They will for short be referred to as 'the first variant' and 'the second variant', respectively.

The paper is organized into five parts 2–6. Part 2 provides an account of an elementary technical framework, in which the two variants are presented. In Parts 3 and 4, the two variants are presented. In Part 5 some attention is given to applications. In Part 6, finally, there is a brief discussion of what to do with records containing unique data.

2. The Technical Framework

2.1. The set of census records

There is one data record for each of the N individuals enumerated in the census. Each record carries data ('values') for Q variables. Some of these variables are qualitative, such as 'sex' and 'marital status', while others are quantitative, such as 'age' and 'income'. The set of census records may be represented by a $N \times Q$ data matrix.

2.2. Data in the public domain

For some of the Q variables, the data about

some individuals are in the public domain. If not already known to the general public, they may be found in registers that the public has access to. Examples of such variables are sex, age, and marital status. The number of variables whose data are in the public domain may be different for different individuals.

The two variants for identifying unique records will be presented with reference to a fixed set of P variables. Data for this set of variables may be in the public domain for a non-negligeable proportion of the census population. We will denote these variables by X, $Y, \ldots$. Occasionally, X, $Y, \ldots$ will also denote the values of these variables (i.e., the data). For short, we will refer to these data as 'the P-set' of data. Throughout the rest of this paper, the variables/data will be listed in the order just presented. For the remaining Q-P variables, the data are not in the public domain; these are the data that would be the prime target of a potential invader of privacy.

The data in the P-set are all assumed to be categorical, with the categories denoted by '0', '1' ... . 'C-1', where C, the number of categories, may be different for different variables. If that is the case, the number of categories will be denoted by C_X, $C_Y, \ldots$. The categories for each variable must be chosen in a way that does not facilitate efforts to link records with individuals. In the following section this goal will be explained.

2.3. *Unique P-sets of data*

The problem addressed in this paper is that there may be P-sets of data that are unique. A P-set of data is defined as unique in the set of N records in the following two cases:

i) there is but one individual with that specific set of data; or
ii) there are a small number k of individuals, say $k = 2$ or 3, with that specific set of data.

In the first case, access to the P-set of data makes it possible for anyone to identify the corresponding individual and, with access to the N records, learn about his or her data for the remaining Q-P variables, even if the records are de-identified. In the second case, $k - 1$ of the individuals may, in collusion with each other, identify the record of the remaining individual. In what follows, we will, for simplicity, relate the presentation to the case with $k = 2$. In both cases, the P-set of data serves as a 'quasi-identifier'.

Clearly, there may be two or more different P-sets of data that are unique. The two variants to be discussed in Parts 3 and 4 handle these situations.

The desideratum suggested at the end of Section 2.2. will now be discussed with reference to the variable 'place of residence', which typically would be one of the variables with data in the P-set. The categories may refer to the sizes of the places of residence. If one of the categories consists of places with small populations, a potential intruder may find it feasible to search for records with a specific P-set of data, for individuals who live in the small place of interest to him. If he finds one such record, it would certainly be unique for that area. To counteract such a search, categories should consist of places with reasonably large populations only.

2.4. *Simplification of the account*

We present the two variants for identifying records with unique P-sets of data in the following simple contexts.

2.4.1. Use of miniature data matrices

Through much of the discussion, we will refer to data matrices with $N = 7$ records with data in the public domain for $P = 5$ variables. This will facilitate the use of numerical illustrations.

2.4.2. Number of categories

For all P variables, the number of categories will be assumed to be at most 10. This means

that a 1-digit number ('0', '1', ... '9') suffices to record category.

2.4.3. Representation of the *P*-set of data

For the *j*th record, $j = 1 \ldots N$, the *P*-set of data may be written as a row vector:

$$R_j = (X_j, Y_j, Z_j, U_j, V_j)$$

which in a specific instance would appear as:

$$R_j = (0, 1, 3, 2, 6).$$

Alternatively, the *P*-set of data may be written as a 5-digit integer. For the specific instance just presented, this integer would be:

$$I_j = 01326.$$

In what follows, we make use of both representations.

3. The First Variant

3.1. The basic idea

The N data records are represented by N integers as discussed in Section 2.4.3. These integers are sorted 'by size'.

We consider two cases:

i) the P variables have the same number C_0 of categories, the special case; and
ii) the P variables do not have the same number of categories, the general case.

3.2. The special case with $C_0 = 2$ for all variables

Consider the following data/integers:

Record No.	Data				
1	1	0	1	1	0
2	0	1	1	0	1
3	1	0	1	1	0
4	0	1	1	0	1
5	1	1	1	0	1
6	0	1	1	0	1
7	0	1	1	0	1

While the number of different possible integers is $2^5 = 32$, there are, of course, for a matrix with $N = 7$ records at most 7 different integers. We will consider three different approaches to sorting the records.

3.2.1. Approach no. 1

According to this approach, the 7 integers (and hence the 7 records) are sorted into increasing (or decreasing) order. The outcome would be:

Record No.	Data				
2	0	1	1	0	1
4	0	1	1	0	1
6	0	1	1	0	1
7	0	1	1	0	1
1	1	0	1	1	0
3	1	0	1	1	0
5	1	1	1	0	1

The table shows that:

i) record no. 5 is unique: there is only one such record; and
ii) there are two records, no. 1 and no. 3, with the same *P*-set of data; hence these records are also unique.

3.2.2. Approach no. 2

The records are processed in the order they appear in the data matrix. This approach works as follows.

Processing record no. 1 corresponds to the following outcome:

Record No.	Data	# Records
1	1 0 1 1 0	1

This outcome is 'updated' when record no. 2 is processed:

Record No.	Data	# Records
1	1 0 1 1 0	1
2	0 1 1 0 1	1

This outcome in turn is 'updated', when record no. 3 is processed:

Record No.	Data	# Records
1,3	1 0 1 1 0	2
2	0 1 1 0 1	1

and so on. The final outcome would be:

Record No.	Data	# Records
1,3	1 0 1 1 0	2
2,4,6,7	0 1 1 0 1	4
5	1 1 1 0 1	1

which, of course, is identical with the outcome when using approach no. 1.

3.2.3. Approach no. 3

With $2^5 = 32$ different possible integers, a catalogue of all 32 possible integers would be constructed:

Integer No.	Data
1	0 0 0 0 0
2	0 0 0 0 1
.	
.	
31	1 1 1 1 0
32	1 1 1 1 1

The records are now sorted into these 32 categories of integers. The outcome would be recorded as follows:

Data	Record No.	# Records
0 0 0 0 0	–	0
0 0 0 0 1	–	0
.		
.		
0 1 1 0 1	2, 4, 6, 7	4
.		
.		
1 0 1 1 0	1, 3	2
.		
.		
1 1 1 0 1	5	1
.		
.		
1 1 1 1 0	–	0
1 1 1 1 1	–	0

3.3. The special case with $C_0>2$ for all variables

It is immediately obvious that the discussion in Section 3.2 is directly applicable to the case where all P variables have the same number $C_0>2$ of categories.

3.4. The general case

Consider the following numbers of categories:

X has $C_X = 2$ categories: 0,1
Y has $C_Y = 4$ categories: 0,1,2,3
Z has $C_Z = 3$ categories: 0,1,2
U has $C_U = 2$ categories: 0,1
V has $C_V = 2$ categories: 0,1

and the following data matrix:

Record No.	Data
1	1 3 2 0 1
2	0 2 2 1 0
3	1 1 0 0 1
4	1 3 2 0 1
5	0 2 2 1 0
6	0 2 2 1 0
7	1 3 2 0 1

The approaches no. 1–3 are all directly applicable. The outcome would be as follows. There are three different sets of data:

Data	Record No.	# Records
0 2 2 1 0	2,5,6	3
1 1 0 0 1	3	1
1 3 2 0 1	1,4,7	3

Thus, there is one unique record, viz. record no. 3.

4. The Second Variant

The P-set of data is subjected to a transformation that creates a one-to-one association between the P-set of data ('the original data') and the sums of the transformed data. For each specific sum Σ, there is one and only one

P-set of original data. We will consider two types of such transformations, T_1 and T_2.

4.1. The T_1 transformation

4.1.1. The transformation

The T_1 transformation replaces the original data (X, Y, ...) with numbers (X', Y', ...) selected from the following sequence:

0, 1, 2, 4, 8, 16, 32, ...

We use a scheme that conforms to the numerical sequence described above. This scheme exploits the fact that:

$0+1 < 2$
$0+1+2 < 4$
$0+1+2+4 < 8$
$0+1+2+4+8 < 16$

etc. In Sections 4.1.2 – 4.1.3, the scheme is exemplified.

4.1.2. The special case with $C_0 = 2$ categories for all variables

As in Part 3, we consider a data matrix with data for $P = 5$ variables X, Y, Z, U, V. The original data are transformed according to the following table:

If the original data are:	The transformed data are: X'	Y'	Z'	U'	V'
0	0	0	0	0	0
1	1	2	4	8	16

The transformation proceeds from left to right. Alternatively, it could proceed from right to left.

We apply this type of transformation to the data matrix considered in Section 3.2. The outcome is as follows:

Record No.	Original Data X	Y	Z	U	V	Transformed Data X'	Y'	Z'	U'	V'	Σ
1	1	0	1	1	0	1	0	4	8	0	13
2	0	1	1	0	1	0	2	4	0	16	22
3	1	0	1	1	0	1	0	4	8	0	13
4	0	1	1	0	1	0	2	4	0	16	22
5	1	1	1	0	1	1	2	4	0	16	23
6	0	1	1	0	1	0	2	4	0	16	22
7	0	1	1	0	1	0	2	4	0	16	22

As scanning this table shows, $\Sigma = 23$ corresponds to one record, viz. record no. 5. Moreover, there are two records (no. 1 and no. 3) with $\Sigma = 13$. These three records are thus unique.

4.1.3. The general case

We consider the case discussed in Section 3.4, where:

X has $C_X = 2$ categories, denoted by 0,1
Y has $C_Y = 4$ categories, denoted by 0,1,2,3
Z has $C_Z = 3$ categories, denoted by 0,1,2
U has $C_U = 2$ categories, denoted by 0,1
V has $C_V = 2$ categories, denoted by 0,1

The scheme presented in Section 4.1.2 may be adapted to this case. Proceeding from left to right, the scheme now becomes:

If the original data are:	The transformed data are: X'	Y'	Z'	U'	V'
0	0	0	0	0	0
1	1	2	16	64	128
2	*	4	32	*	*
3	*	8	*	*	*

The outcome is as follows:

Record No.	Original Data					Transformed Data					Σ
	X	Y	Z	U	V	X'	Y'	Z'	U'	V'	
1	1	3	2	0	1	1	8	32	0	128	169
2	0	2	2	1	0	0	4	32	64	0	100
3	1	1	0	0	1	1	2	0	0	128	131
4	1	3	2	0	1	1	8	32	0	128	169
5	0	2	2	1	0	0	4	32	64	0	100
6	0	2	2	1	0	0	4	32	64	0	100
7	1	3	2	0	1	1	8	32	0	128	169

Scanning this table shows that record no. 3 is unique: it is the only record with $\Sigma = 131$.

4.2. The T_2 transformation

4.2.1. The transformation

The following general formulas are used to carry out the transformation:

$X' = 1 \times X$
$Y' = C_X \times Y$
$Z' = C_X \times C_Y \times Z$
$U' = C_X \times C_Y \times C_Z \times U$
$V' = C_X \times C_Y \times C_Z \times C_U \times V$

4.2.2. The special case with $C_0 = 2$ categories for all variables

In this situation, the transformation becomes:

$X' = 1 \times X$
$Y' = 2 \times Y$
$Z' = 2 \times 2 \times Z$
$U' = 2 \times 2 \times 2 \times U$
$V' = 2 \times 2 \times 2 \times 2 \times V$

This yields the same transformed data as the T_1 transformation does.

4.2.3. The general case

When applying the T_2 transformation to the data presented in Section 4.1.3, we obtain:

If the original data are:	The transformed data are: X'	Y'	Z'	U'	V'
0	0	0	0	0	0
1	1	2	8	24	48
2	*	4	16	*	*
3	*	6	*	*	*

We notice that the transformed data are considerably smaller than those arrived at when using the T_1 transformation.

5. Some Aspects of Applications

5.1. The scenario

Much of the previous discussion refers to miniature data matrices. While the use of these matrices allows simple presentations of the basic ideas, a price is paid for this simplification. Simplification can disguise practical problems likely to be present in realistic applications.

In this part, we will sketch a real life scenario with the following features:

i) the data are recorded on magnetic tape for processing on a computer of medium size;
ii) the total number N of records is 'large', say in the order of 10000000;
iii) the number P of variables, for which the data are in the public domain, is 'large', possibly more than 20; and
iv) for one or more of these variables, the number of categories may exceed 10.

5.2. The de-identification of the records

This operation should not be undertaken before the implementation of any identification variant. Having access to the formal identi-

fiers (name, address and/or population registration number) may simplify 'pulling out' the corresponding census forms, if that is what we want to do.

5.3. Choice of approach

To make a rational choice between the two variants for identification of unique P-sets of data, it is desirable to have a cost function that relates the cost of identification to the number of records, the number of variables for the data that are in the public domain and the computer memory necessary to perform each of the variants.

Not having a cost function, we will be satisfied by illustrating a few ways of ensuring efficiency. The illustration refers to use of the first variant, but is also valid for use of the second variant.

Consider the first eight records of a large data matrix, for which $P = 5$:

Record No.	Data				
1	1	1	3	2	2
2	0	0	5	0	2
3	1	1	3	2	2
4	0	0	2	1	1
5	1	1	3	2	1
6	0	0	5	0	2
7	0	0	2	1	1
8	1	1	3	2	2

The records are processed in the order listed, and the number of records is counted for each different P-set of data. Processing record no. 1 corresponds to the following outcome:

Record No.	Data	# Records
1	1 1 3 2 2	1

This outcome is updated, when record no. 2 is processed:

Record No.	Data	# Records
1	1 1 3 2 2	1
2	0 0 5 0 2	1

since the first two records are different. Processing the third record yields the outcome:

Record No.	Data	# Records
1,3	1 1 3 2 2	2
2	0 0 5 0 2	1

When the first eight records have been processed, the outcome is:

Record No.	Data	# Records
1,3,8	1 1 3 2 2	3
2,6	0 0 5 0 2	2
4,7	0 0 2 1 1	2
5	1 1 3 2 1	1

Since there are obviously (at least) 3 records with data

1 1 3 2 2

keeping track of these records is no longer necessary (assuming $k = 2$). Not doing so serves to 'stretch' the memory of the computer.

5.4. Choice of transformation

Given that we have chosen the second variant for identifying unique P-sets of data, a choice still remains between the two transformations, T_1 and T_2.

If P is 'reasonably small', say less than 10, and the same holds true about the number of categories for the variables, this choice would make no significant difference with respect to efficiency. If the above condition does not apply, the T_2 transformation would be preferable, by virtue of its demanding less memory space.

6. Taking Protective Action on the Unique Records

6.1. The final step

Identifying the unique records is the first step in addressing the problem that such records present. The second and final step is to take some action to protect the unique records identified. In Sections 6.2 – 6.5 we will present four possible actions that may be taken.

6.2. Destruction of the records with unique P-sets of data

This is the simple action. If the number of records with unique *P*-sets of data is 'small' relative to the total number of records, and the same holds true for the major domains of study ('breakdowns') of interest, this action may be viable. It has, however, an obvious shortcoming: it will inevitably introduce differences between the statistics based on the full set of records and statistics based on the reduced set of records.

6.3. Blanking out the data in the P-set

If all data in the *P*-set are blanked out, satisfactory protection would be provided against identification. There would be no 'quasi-identifiers'. If, however, some but not all such data are blanked out, the protection may not be as satisfactory as when all data are blanked out. Such a situation may invite a potential intruder to try to link a record with the corresponding individual by means of techniques that make it possible to compromise a database. Two pertinent references for such techniques are Denning and Schlörer (1980) and Dobkin et al. (1979).

6.4. Data perturbation

This action calls for replacing the original data with different (new) data for one or more variables concerned. Techniques for rounding counts in contingency tables could be applied. A pertinent reference is Cox (1983).

6.5. Encryption of the data

This entails a *reversible* transformation of the data. This reversibility obviously has an advantage over the data perturbation scheme. Encryption may, however, prove more expensive.

7. Acknowledgements

In preparing this paper, I received helpful comments and suggestions from Fil. Kand. Johan Berglind, Department of Mathematical Statistics, Lund University, participants of a seminar in the Department of Statistics, the University of Stockholm and Professor Edmund A. Lamagna, Department of Computer Science and Statistics, University of Rhode Island.

8. References

Cox, L. (1983): Some Mathematical Problems Arising from Confidentiality Concerns. Statistical Review, No. 5, pp. 179-189.

Denning, D.E. and Schlörer, J. (1980): A Fast Procedure for Finding a Tracker in a Statistical Database. ACM Transactions on Database Systems, Vol. 5, No. 1, pp. 88-102.

Dobkin, D., Jones, A.K., and Lipton, R.J. (1979): Secure Databases: Protection Against User Influence. ACM Transactions on Database Systems, Vol. 4, No. 1, pp. 97-106.

Received August 1986
Revised September 1986