Estudios demográficos y urbanos, vol. 31, 3, núm. 93, 2016, pp. 811-832.

El formato Redatam - english version.

De Grande, Pablo.

Notes and comments

The Redatam format*

English translation of the article:
De Grande, P. (2016). El formato Redatam. *Estudios Demográficos y Urbanos*, 31(3), 811–832. https://doi.org/10.24201/edu.v31i3.15

Pablo De Grande**

The Redatam statistical package is a software developed by ECLAC and widely used in the countries of the Americas for the dissemination of census statistics. Although it is free to use, its code is not open source and the structure of the format used to store the information is not public. This article presents the results of a research project on the data structure of this tool. Among them, the following stand out: to a) *a preliminary specification of the Redatam format,* b)*the publication of a tool for exporting Redatam databases, and* c)*evidence that, contrary to what is established in the technical documentation, the software does not implement compression and encryption strategies for the microdata stored by it.*

Keywords: open access; ECLAC; Redatam; statistical analysis; confidentiality.
Date of receipt: July 6, 2015.
Date of acceptance: February 16, 2016.

Abstract

The Redatam statistical package is a software package developed by ECLAC and widely used in countries of America for the dissemination of census statistics. Although it is free to use, it is licensed as proprietary software (not open source) and stores its data in a non-public format. This article introduces research results describing the data structure used by this software. They include: a) a preliminary specification of the Redatam format, b) a tool for accessing and exporting your databases, andc)the evidence that –contrary to what the technical documentation states– Redatam does not implement strategies for compression and encryption of the microdata it stores.

Keywords: open access; ECLAC; Redatam; statistical analysis; confidentiality.

* I am grteful for the sensitive and helpful contributions in the preparation of this publication from Alejandra Russo, Mariana Bordoni and the peer reviewers. I would also like to thank Agustín Salvia for his trust and support in guiding me so patiently in my interest in investigating census data and other statistical problems.

* * University of Salvador, Institute for Research in Social Sciences (Idicso).
Postal address: Pte. Perón 1818, 3rd floor, City of Buenos Aires (C1089AAU), Argentina.
Email: pablodg@gmail.com

Introduction

With the increasing availability of computer resources for the circulation of large databases, the tension between two fundamental rights has become a challenge in the dissemination of census results. On the one hand, the protection of the privacy of the data provided by individuals suggests that only a small number of technical staff should be able to consult the data obtained, under strict terms of confidentiality. On the other hand, the social relevance of the dimensions studied encourages the defence of the right to full access to such statistical information for analysis and discussion.

This is the problem that gave rise to Redatam, a package developed by ECLAC to make the circulation of census data compatible with the protection of the confidentiality of personal data that could potentially be contained therein.

Redatam is currently a software for the distribution and exploitation of demographic data widely used in countries in the Americas. It was created by Serge Poulard at the Latin American and Caribbean Demographic Centre (CELADE), which is part of the United Nations Economic Commission for Latin America (ECLAC). CELADE is also responsible for its maintenance and distribution, organizes courses and distributes material and new versions periodically.

Dating back to 1986, Redatam has become a “de facto standard” for publishing census results over the last two decades. Thus, Argentina, Colombia, Chile, Mexico and Peru, among others, have adopted this tool to disseminate their census databases, both via the web and in the form of a desktop application for Windows.

Redatam's usage mode allows its users to calculate totals and percentages based on the available variables. In this way, for example, it is possible to consult the number of households in a town or province, or the number of unemployed people by sex and age.[1]

This tool has represented a very significant advance in the capabilities of users in a variety of research centers.

[1] Through a specific syntax, the software allows to build variables *ad hoc* at any level (for example, locality, household, person), giving flexibility of consultation with the irremovable condition that the outputs are simple counts (or their direct percentages).

tion and state agencies to perform dynamic tabulations with census data. Given the restrictions of statistical offices to provide primary data from their censuses, Redatam was a superior proposal for the distribution of statistical material. In this sense, it was sufficiently conservative to be accepted by the producers of the information, and sufficiently powerful to be adopted (with the training needs that this entailed) by researchers and technical staff to the extent that the databases were available.

Although Redatam is a public software, which can be downloaded and installed free of charge, it should be noted that its source code has not been made available to the academic community (i.e., it is not open source). Furthermore, the format used to host the data has not been documented by ECLAC or other organizations. This article presents research results related to the latter of the two aspects mentioned, that is, the closed nature of the data format.

In order to ensure greater transparency in the research processes and a better analytical capacity of the scientific community on the available demographic data, the goal was to analyse the format in which Redatam stores information. This was to achieve two objectives: on the one hand, to evaluate the level of reliability in the protection of the data offered by the software, while at the same time – if it was possible to decode the way in which the information was stored – to allow more complex and dynamic forms of analysis for the available data.

As a result of this work, a partial specification of the data structure used in the Redatam database distribution has been reached. This specification also allowed the development of an open source tool for the comprehensive reading of databases in Redatam format (available at https://www.aacademica.org/conversor.redatam/).

In the following section, some limitations of the Redatam package are discussed, highlighting its barriers to the statistical exploitation of information and the absence of specifications regarding the protection it provides for distributed data. Secondly, in the Methodology section, the inputs and procedures with which the analysis was performed are indicated. Thirdly, in the Results section, the data structure inferred from the Redatam databases is detailed. Finally, in the Discussion and Conclusions sections, the main findings summarize and put into context, reconsidering where the balance between confidentiality and access stands after the use of Redatam in a large part of the 2000-2010 census series in America.

Difficulties in statistical analysis and the evaluation of the protection of information

This research began with the question of how it was possible that there was so much valuable information in Redatam format and it could not be processed statistically in complex ways. Redatam has been offering an application and query syntax for extracting totals and percentages for almost twenty years, with no options to extend this calculation capacity.

Since data can only be viewed as simple tabulations, their use for more advanced statistical operations is severely hampered. Performing an analysis of variance, calculating the confidence of a difference in means, or performing inferential models requires extensive work, such as *artisanal* data extraction by calculating the totals for all combinations of categories of the variables involved, then partially reconstructing a working base with them[2]. This type of use, even in cases where it is possible, requires advanced skills in Redatam, makes exploratory analysis difficult, and requires large amounts of time to resolve operations that are basic when the data is in the form of rows in a table.

At this point, the problem is not so much the fact that Redatam does not perform more complex operations, but rather the fact that it cannot be extended by third parties or interact with other statistical packages. The ways for this type of interaction could be diverse, but it is worth highlighting at least three that are widely used in system integration:

1) Extensibility interfaces: It is common for software packages to offer channels for adding externally programmed modules that interact with the main application. For example, the geographic information program ArcGis allows, through scripts in various

[2] This strategy was used in De Grande and Salvia, 2008.

languages, to access and modify the data layers of their maps; the case of macros in Microsoft Office, the plugins in web browsers or applications in operating systems (traditional and mobile) are cases of successful extensibility through this means.

2) Openness of the data format: through controlled documentation of the versions of the format in which the data is saved, it is possible to give other software providers or researchers and research teams the possibility of making their applications compatible with the format itself. In the case of Redatam, it only uses known formats for exporting the results of the tabulations. The Acrobat PDF format is a success story of growth through a publicly specified format.

3) Open source code: Open availability of the code of a software package allows other programmers to examine the instructions that form part of a program, making contributions or improvements to it. Knowing the internal mechanisms of an application also often allows those who can read the language in which it was written to clear up doubts and learn about the detailed behaviour of the program in question. The Linux operating system and the R+ statistical package are two successful cases of extensibility through open source code.

Any of these three paths could allow the Redatam user community to grow towards more advanced forms of analysis on the data currently available.

A second starting point for this research was represented by the question of how protected the data were in a Redatam database. The main statistical offices have distributed their data in Redatam, rather than in more familiar formats (such as DBF tables or SPSS databases), possibly trusting that it was an effective way of safeguarding the confidentiality of primary data.

In this sense, ECLAC presents Redatam as a package that protects microdata by encrypting it (ECLAC, 2015), thus preventing people other than the producers of the information from accessing it. The introduction to the Redatam documentation states:

> *Population and housing censuses, agricultural censuses, household surveys, vital records, etc., are databases containing millions of records on dwellings, households and people. These data, organized hierarchically in a Redatam format, are stored in encrypted and highly compressed form, thus protecting the statistical confidentiality of the person themselves (ECLAC, 2015).*

Similarly, this aspect was highlighted in the launch of its 2002 version, stating that "external databases are converted to Redatam's own format, which compresses, encrypts and inverts the original data in order to combine efficiency with the confidentiality of the information" (Faijer and Poulard, 2002: 326).

But how does Redatam encrypt information? Cryptography is a specific discipline, which has gained enormous popularity in the last thirty years (Katz and Lindell, 2007) with the development of protocols to protect internet connections, banking operations, documents, personal signatures and emails, among others. However, Redatam's technical documentation does not give any clues about the type of encryption it performs on information. Similarly, to date no records have been found of verifications on the strength of this last aspect by the academic community or the statistical institutes that use it.

As this is a package aimed at potentially confidential data, this research sought to provide clarity on this aspect, thus allowing national statistical offices and the Redatam user community in general to make an informed decision on which columns to incorporate or not in the databases, given the reliability of the safeguards offered.

Methodology

To analyze the Redatam storage scheme, a set of public databases in this format was used, as well as the Redatam package in its desktop version for Windows R+SP V5. This version has the capacity to access and create databases, allowing the user to play the role of both statistics consumer and database producer. It can be downloaded publicly from the ECLAC website.

The analysis was carried out using three strategies deployed in parallel: on the one hand, typical reverse engineering criteria were followed to investigate unknown formats, observing variations in simple files (Eilam, 2005: 200); on the other, samples of existing databases in circulation were analyzed; finally, a tool was generated to validate the hypothesis under construction aimed at reconstructing the original microdata sets.

In the first strategy, groups of elementary files were produced and their variations were examined. This meant starting with the creation of a database with only a table with one row and one integer column. Then an additional variable of the same type was added. Then the data type was modified, and so on, observing the changes produced by the tool in the databases.

To carry out the second strategy, a corpus of pre-existing databases was defined to be used as a reference. It was mainly created from the databases available in Redatam and SPSS formats on the website of the Institute of Statistics and Census of Argentina[3] .The selection of these control databases was aimed at validating what was observed in small databases from databases *royal*, generated at different times and under different needs. They were also used to qualitatively observe the salient features of the data structure investigated, such as the typical number of files, the extensions used or the general file sizes.

In order to be able to verify the findings produced for the description of the format in a fast and massive manner during the course of the research, as a third methodological strategy a tool was developed that would implement these definitions and apply them in the reconstruction of the microdata contained in the Redatam databases. This tool was named Redatam Converter, and is available in open source for evaluation and experimental use in the GitHub repository[4]. The application currently has the ability to export the structure and microdata from Redatam databases to SPSS files (.sav) or plain text files (.csv). External users who downloaded the application reported having successfully converted census databases from Argentina, Bolivia, Chile and Uruguay.

[3] http://www.indec.gov.ar/bases-de-datos.asp
[4] https://github.com/discontinuos/redatam-converter

Results

As indicated above, the analysis carried out has advanced to the point of having a partial but sufficient specification for the full reading of the microdata of a Redatam database. This section presents the recognized file and data structure, specifying the function of each file type and its internal structure.

First, it was possible to recognize that Redatam's databases were organized from a "dictionary" file, which contained the list of entities and variables and their definition. In addition to the dictionary, there were also data files (where the values for each row of each variable were found) and correspondence files (where the relationship between entities at different levels is indicated, such as which countries each province corresponds to, or which household each person corresponds to in a database).

The types of data identified are described below, followed by the file types in which they were contained in the databases analyzed.

Data types

Within the framework of Redatam storage format recognition, the data variants used by the software were examined.

In the case of text values, it could be observed that Redatam stores variable-size strings in the dictionary description (which we will call the STRING type here)[5] and fixed-size (here we will call it the CHAR type) in data files. In both cases, characters are stored using the Windows-1252 (8-bit) code table.

In the case of numeric values with decimals, Redatam stores eight-byte floating point values for persistence (a type we will call DOUBLE). For integer values, it uses a set of variable data types depending on the range of values to be stored (which we will call INT16, INT32, and BITS(n) types).

These data types are specified in Table 1 and are used in subsequent descriptions to indicate the storage methods for each value.

[5] In some cases the names of the types are derived from the denomination used in Redatam; in others a name was assigned *ad hoc*, seeking to use terms commonly used in the specification of data structures for packages or computer languages.

TABLE 1

Types of data used in the description

Data type	*Description*	*Example*
BITS(n)	Stores arbitrarily sized bit sequences to hold integers. BITS field values are retrieved by reading INT32 integers, so a series of BITS values will always be a multiple of 4 bytes in size.	0xA0860100 => 11000011010100000 => BITS(4) => 12; 3; 5; 0
BYTE	1-byte unsigned integer. Variable-size	0x02
BYTE[]	sequence of bytes.	0x02050202040405
CHAR(n)	Fixed-size character sequence. Like the STRING type, special characters are encoded using the Windows default character table, or Windows-1252.	0x504552524F => DOG
DOUBLE	Floating point number, stored following the IEEE 754 standard used by most programming languages.	0x547424971F88B340 => 5000.1234
INT16	2-byte unsigned integer. 4-byte	0x0401 => 260
INT32	unsigned integer.	0xA0860100 => 100,000
STRING	Stores variable-size text strings. It presents 2 bytes at the beginning describing the size of the contained text, after which the text itself is found. If it is necessary to store strings equal to or longer than 65,535 characters (the maximum size specifiable in 2 bytes) it indicates the value 65,535 in the first 2 bytes and then reserves a 4-byte integer to describe the length of the long text.	0x43415341 => HOME
Note: In all cases where values greater than 1 byte are stored, the storage mode is*little endian*, that is, the smallest byte is stored first.		0x0204 => 0x04; 0x02

Source: Prepared by the authors based on analysis of archives.

Dictionary file

As for the dictionary file, it was found that it stores the list of entities that make up the database, including the details of variables and labels for each of them. The Redatam data schema assumes the existence of hierarchical data, that is, a universe of data in which the entities are related as *parent-child.* Typically in census structures this relationship takes the form of a sequence whose highest level is the country, the next level is the province or state, the next are the departments, districts or localities, following intermediate levels until reaching the housing, household and person levels.

The file structure has a header containing general database attributes, which has not been described at this stage of the research because it is not binding for the description of the data. Following the header is a list of blocks that describe each of the types of entities contained in the database (for example, provinces, departments, households, persons).

Each entity block, in turn, is decomposed from an initial list of entity attributes (such as its name, its parent entity, the name of its identifier variable), followed by a list of descriptor blocks for each variable that the entity possesses. Each variable block in turn includes attributes of the entity, which indicate the data type, name, extended description (its label), and labels for the possible values of the variable, among other elements. A detailed description of these structures can be found in Table 2.

Correspondence file

The observation also revealed that the .PTR files (which we have called "correspondence" files here) function as indexes or reference tables to determine which entity at a higher level corresponds to an entity at a lower level. There is a correspondence file for each type of entity contained in the database. These allow us to determine, for example, when calculating a result, which province a certain department is in, or which household a certain person is in.

TABLE 2
Descriptive sheet for the “dictionary” file type

	Description		
File type	Dictionary		
Extension	DEC		
Specification Level	Partial		
Object	Contains the list of entities and their variables (columns).		
	Structure		
Field	*Content*	*Description*	*Example*
Header	BYTE[]	Unknown. It gathers a group of data that precedes the entities and that was not analyzed because it did not appear to be necessary for reading the data.	
Entities	Sequence of entities	Following the header are entries describing the entities that form part of the database.	
Name1	STRING	Name of the entity.	DEPT
Name2	STRING	Repeats the previous value. Omitted if the entity has no parent (top level).	DEPT
Father	STRING	Name of the parent entity with respect to the current one. Empty STRING if it is the parent entity.	PROV
Description	STRING	Extended description of the entity.	Department
Correspondence file	STRING	Details which file describes the entity's mappings to its parent entity.	CV100000.ptr
<unknown>	INT16	2 bytes of unidentified use.	

(continued)

TABLE 2

(continued)

Field	*Content*	*Description*	*Example*
Identifier Variable	STRING	Specifies the name of the variable within the entity; holds descriptive codes for each row.	PROVID
Descriptor Variable	STRING	Specifies the name of the variable within the entity; holds textual descriptions of each row.	PROVINCE
<unknown>	INT32	4 bytes of unidentified usage. 1	
<unknown>	BYTE	byte of unidentified usage.	
Number of variables (?)	INT32	Number of variables. This value was not consistent across all databases, so the converter does not use this value.	12
<foot>	BYTE[]	Unknown. End of entity description. The corresponding values were not decoded and are not necessary to extract the information.	
Variables	Sequence of variables	Then there are entries describing each variable of the entity. The beginning of these is recognized by the existence of entries in the form “<variable name> DATASET”	12
Name	STRING	*Variable name*	PROV
Statement	STRING	The declaration is specified after the DATASET prefix. It consists of three elements, separated by spaces. These are: the data type of the variable, the file where the data corresponding to the variable is stored, and the size. For the data type indication, the possible values are: BIN: integer values with a specifiable fixed size stored in 4-byte blocks in binary mode.*big-endian*. CHR: Text values with a specifiable fixed size.	DATASET CHR 'CP200000.rbf' SIZE 2 DATASET BIN 'CP4541.bin' SIZE 7

		DBL: Decimal (floating point) values specified in 8 bytes.to INT: Integer values from 0 to 65,535. LNG: Integer values from 0 to 4,294,967,296. PCK: integer values with specifiable fixed size stored in 4-byte blocks in mode*little endian*. The size is indicated in bytes in the case of CHR variables and in bits in the case of BIN and PCK type variables. Variables of INT, LNG and DBL type are of fixed size, being 2, 4 and 8 bytes respectively.	

Filter	STRING	Indicates whether the variable should be used only under certain conditions.	DWELLING.
			V02 = 1
			AND HOME.
			NHOG = 1

Range	STRING	Minimum and maximum possible values for numeric variables, separated by the term 'TO'.	1 TO 10
Guy	STRING	Type of data stored, indicating whether it is a numeric or text value. Possible values are INTEGER for integers, REAL for numbers with decimals, and STRING for text.	INTEGER
Tags	STRING	The list of labels to use for the variable. Entries are separated by Tabs (character 9), and values are separated from labels by spaces.	1 Male{TAB} 2 Woman

(continued)

TABLE 2

(concludes)

Description	STRING	Extended description of the variable (variable label).	Country of birth
Descriptors	STRING	A list of elements is stored that allows additional aspects of the variable or its values to be described. Attributes are optional and are stored separated by spaces. They are: ALIAS: allows you to define an alternative name for the variable. DECIMALS: number of decimals to display for REAL data types. GROUP: allows you to indicate the name of the group in which the variable should be displayed. MISSING: indicates the value that indicates unrecorded data. NOTAPPLICABLE: indicates the value that indicates non-relevant data.	MISSING 4 NOTE APPLICABLE 0 GROUP EDUCATION ALIAS ALPHABETS

[to]The range of a double-precision (8-byte) data type is -1.79769313486231570E+308 to -4.94065645841246544E-324 for negative values and 4.94065645841246544E-324 to 1.79769313486231570E+308.

Source: Prepared by the authors based on analysis of archives.

The way this is solved is by keeping in these files an ordered list with as many elements as the top-level entity has. Each of these elements contains the number of lower-level entities that correspond to the top-level entity, which are ordered according to this criterion (which is its top entity).

Let us consider an example in which there is a table with 24 provinces, on which another table with 240 departments depends. The correspondence file indicated for the entity “Departments” will contain 24 elements (after a starting value of zero that the file has), specifying in each of them the number of departments that correspond to each province. If the departments were homogeneous in their distribution – that is, if each province had 10 departments in its jurisdiction – the list would be composed of a series of 24 values of 10 (the number of departments in each province). If instead the first province had 15 departments and the second had 5, the content of the correspondence file would start with the number 0, as it always does, then there would be a 15 and then a 5. The details of this structure can be seen in table 3.

Data file

The data files of the analyzed package, indicated in the dictionary for each variable, contain the information about the values that each variable has in each entity. This implies that there is a data file for each variable (for example, Person. Age, Person. Sex, Person. Occupation). For this reason, there is not a single data file for each type of entity (such as Persons), so that querying a list of entities requires reading several files simultaneously.

This strategy may have been adopted to speed up data reading, since in this way Redatam only accesses the data blocks corresponding to the variables selected in each query, avoiding reading the entire entity record. The details of the storage structure are specified in Table 4.

TABLE 3

Descriptive sheet for the “correspondences” file type

	Description		
File type	PTR correspondence file		
Extension			
Specification Level	Complete		
Object	Contains the way in which entities at different levels are related.		
	Structure		
Field	*Content*	*Description*	*Example*
Initial row	INT32	Constant value at zero.	0x00000000
List of rows by entity	INT32 sequence	Presents a sequence of values indicating the number of rows in the child entity that correspond to the parent entity.	
Rows per entity	INT32	Value for the row corresponding to the position in the list.	512

Source: Prepared by the authors based on analysis of archives.

TABLE 4
Descriptive sheet for the “data” file type

	Description
File type	Data file
Extension	RBF. In older databases the extension .BIN may be found. Complete
Specification Level	
Object	Contains the values corresponding to a variable of an entity. The structure depends on the type of data stored.

Structure			
Field	*Size*	*Description*	*Example*
List of values	Sequence of values	Presents a sequence of values that allow reconstructing the content of the variable for each row of the entity. The list will have as many rows as there are elements for the entity.	
*Structure for data type*BIN			
Value	BITS(n)	Integer value of arbitrary size corresponding to the position in the list. BITS value series persist in blocks of 4 bytes, with the highest byte first (order).*little endian*). Older databases use the BIN dataset format, while newer ones use the PCK format.	12; 3; 5; 0.
*Structure for data type*CHR			
Value	CHAR(n)	Fixed-length text value for the row corresponding to the position in the list.	DOG
*Structure for data type*DBL			
Value	DOUBLE	Floating point value for the row corresponding to the position in the list.	5000,1234

(continued)

TABLE 4
(concludes)

Structure			
Field	*Size*	*Description*	*Example*
*Structure for data type*INT			
Value	INT16	Short integer value for the row corresponding to the position in the list.	512
*Structure for data type*LNG			
Value	INT32	Long integer value for the row corresponding to the position in the list.	19772501
*Structure for data type*PCK			
Value	BITS(n)	Integer value of arbitrary size corresponding to the position in the list. BITS value series persist in blocks of 4 bytes, which form an integer in the format*big-endian*((i.e. the bytes with the highest weight are at the end). As with the BIN type, once the 4-byte block has been read, the number of bits corresponding to each successive element is taken.	17; 1; 8; 2.

Source: Prepared by the authors based on analysis of archives.

Discussion

In cryptography and computer security the term 'security through obscurity' refers to the strategy by which protection is sought to be effective by keeping the procedures for ensuring it secret. In contrast to this, there is a consensus in contemporary cryptography regarding the validity of the Kerckhoffs principle, which maintains that in a cryptographic system “nothing should be secret except its key”: that is, to maximize the security of a protection, the operation of its mechanisms must be known (Ferguson, Schneier and Kohno, 2010: 74). Thus, the forms of encryption used for encrypted data exchanges on the Internet (such as the SSL/TSL protocol or the IPSec protocol) are publicly documented and are in a constant process of review and discussion by the community of computer security analysts (Stapleton, 2014).

In the case of Redatam, we have encountered a borderline case of security through obscurity: the confidence that the data storage scheme would remain hidden seems to have led not to an implementation of weak encryption, but to no encryption at all.

In this respect, it should be noted that the results of this exploration were partly unexpected, since the Redatam team had claimed, at least since 2002, that the software worked by compressing and encrypting data. As it was found out, neither of these claims is correct.

Regarding the use of space (compression), it can only be said that Redatam stores data in a standardized manner[6], that is, it stores data without repeating, for example, housing data for each household, or data for each household for each person. In this sense, it behaves like any relational database, storing a table for each type of entity and storing the data according to its size. However, neither in recent bibliography (Román González, 2012) nor in older bibliography (Coello and León, 1994) does normalizing a database specifically constitute a data compression method.

Regarding encryption – and this is perhaps the most problematic aspect – no explicit data protection strategy was found during the analysis. Each record was located at the

[6] For a precise definition of the notion of normalization, see Silberschatz, Korth and Sudarshan, 2002.

stored one below the other, without alterations to the texts or the numbers that represented the values, or to the order of the individual data in each record. From the most rudimentary encryption strategy – such as having a substitution table – to the use of validated algorithms that allow the information to be encrypted or signed, none of this was part of the data consulted in the Redatam databases accessed. Thus, as a consequence of the absence of encryption strategies, the microdata in the Redatam databases can be read directly[7]. Furthermore, as a consequence of the absence of data signing strategies, the data may be modified intentionally or accidentally without Redatam or its users being able to validate it.

Returning to the questions raised at the beginning of this article, it is worth asking how these findings affect the current state of tension between protection and dissemination of census data. As stated above, Redatam has allowed the scientific community to expand its available capacity for analysis of census microdata by producing a general publication of databases. However, after twenty years of progress in this direction, we find ourselves at a juncture that places significant limits on this strategy: on the one hand – with the facilitation of the use of advanced statistical techniques – the Redatam software is not as flexible as many of its users require. On the other hand, it is no longer possible to affirm that the Redatam package protects microdata as has been maintained up to now: it is possible, in a trivial manner, to convert a Redatam database into lists of households and people in standard database formats. Both facts suggest the need to review the policies for publication and distribution of statistical information in view of future censuses.

[7] It should be noted here that while it is a significant problem that software advertises capabilities that it does not deploy, the safeguarding of individual privacy is largely covered by the fact that statistical institutes remove from their databases the columns that involve personal data such as names, telephone numbers and addresses of individuals before converting them to the Redatam format. One country that adopts this approach as a policy is Uruguay, which distributes its census databases at the microdata level publicly (in DBF and SPSS formats), considering them sufficiently anonymous to allow their dissemination.

Conclusions

In summary, progress has been made towards a preliminary specification of the Redatam format. The need to make research processes transparent, including the circulation and use of statistical information, has been highlighted.

As part of this research, a portable, extensible and open source tool was produced (De Grande, 2015) that allows the validation of assumptions regarding the Redatam format. This tool has been able to successfully read and export all the databases evaluated to date. The export of data in Redatam format emerges as a crucial step for an in-depth analysis of the available census information and the real situation regarding the balance between accessibility and confidentiality.

References

ECLAC (2015),*R+SP Process Basic Tutoring*, Santiago de Chile, Eco-Commission Economics for Latin America and the Caribbean http://www.redatam.org/ cdr/ Tutoriales/Process_Esp.html (June 30, 2015).

Coello, C. and H. Hernández de León (1994), "Database compression", *Proceedings of the VIII International Symposium on Computer Applications*, Antofagasta, November 21-25, pp. 87-94.

De Grande, P. (2015),*Redatam converter (software)*, Buenos Aires, Discontinuous. Available at: http://www.aacademica.org/conversor.redatam (January 13, 2016).

De Grande, P. and A. Salvia (2008), "Segregación residencial socioeconómica y espacio social: deserción escolar de los jóvenes en el área metropolitana de Gran Buenos Aires", in Agustín Salvia (comp.), *Jóvenes promesas. Trabajo, educación y exclusión social de jóvenes pobres en la Argentina*, Buenos Aires, Miño and Dávila. Available at https://aacademica.org/pablo.de.grande/5 (April 12, 2015).

Faijer, D. and S. Poulard (2002), "REDATAM software for dissemination and analysis "Census data analysis",*Population Notes*, vol. 75, pp. 321–341. Available at: http://repositorio.cepal.org/bitstream/handle/11362/12742/ np75321341_es.pdf?sequence=1 (18 May 2015).

Eilam, E. (2005),*Reversing: secrets of reverse engineering*, Indianapolis, Wiley.

Ferguson, N., B. Schneier and T. Kohno (2010),*C ryptography Engineering. Design Principles and Practical Applications*, Indianapolis, Wiley Publishing. Katz, J. and Y. Lindell (2007). *Introduction to Modern Cryptography: Principles and Protocols,* Boca Raton, CRC Press.

Román González, A. (2012), "Data classification based on compression", *ECIPerú Magazine*, vol. 9, no. 1, pp. 69-74. Available at: https://hal.archives-ouvertes.fr/hal-00697873/document (May 18, 2015). Silberschatz, A., H. Korth and S. Sudarshan (2002),*Database Fundamentals*,

Madrid, McGraw-Hill.

Stapleton, J. (2014), *Security without Obscurity. A Guide to Confidentiality, Authencertification, and integrity*, Boca Raton, CRC Press.