μ ARGUS

version 4.3

User's Manual

Statistics Netherlands
P.O. Box 24500
2490 HA The Hague
The Netherlands
email: aj.hundepool@cbs.nl

Project: ESSnet-project
Date: January 2010
BPA no: 768-02-TMO

Contributors: Anco Hundepool, Aad van de Wetering and Ramya Ramaswamy
Luisa Franconi. Silvia Polettini and Alessandra Capobianchi (Risk models)

Peter-Paul de Wolf (PRAM), Josep Domingo and Vicenc Torra (Numerical micro aggregation, rank swapping and synthetic data), Sarah Giessing (Record linkage)

Contents

1 Preface....5
1.1 About the name ARGUS....5
1.2 Contact....6
1.3 Acknowledgements....6
1.3.1 The CASC-project....6
1.3.2 The CENEX project....8
1.3.3 The ESSNet project....9
2 Introduction....10
2.1 System Requirements....10
3 Producing safe microdata....11
3.1 Safe and unsafe microdata....11
3.2 Statistical disclosure control measures....13
3.2.1 Global recoding....13
3.2.2 Local suppression....14
3.2.3 Top and bottom coding....14
3.2.4 The Post RAndomisation Method (PRAM)....15
3.2.5 Numerical Micro Aggregation....16
3.2.6 Multivariate fixed-size microaggregation....17
3.2.7 Numerical Rank Swapping....18
3.2.8 Synthetic Data....18
3.2.9 Record Linkage....21
3.3 The Risk approach....22
3.3.1 Introduction....22
3.3.2 Disclosure and disclosure scenario....24
3.3.3 Measures of risk....24
3.3.4 Releasing files with a hierarchical structure....32
3.3.5 Some Comments on Risk Estimation....34
3.4 A guided tour of μ-Argus for the application of the Individual Risk Methodology....35
3.4.1 Specification of a household structure in the Menu Specify|Metadata....37
3.4.2 Menu Specify|Combinations....39
3.4.3 Menu Modify|Risk Specification: the Risk Chart window....43
3.4.4 Menu Modify|Household Risk Specification: the Household Risk Chart window....48
3.4.5 Output: Make protected file window....50
3.5 Information loss....53
3.6 Sampling weights....53
3.7 Household variables....54
3.8 Blocking procedures....54
3.9 Functional design of μ-ARGUS....55
4 A tour of μ-ARGUS....57
4.1 Preparation....57
4.1.1 Specify combinations....59
4.2 The process of Disclosure Control....61
4.2.1 Global Recoding....62
4.2.2 PRAM specification....64
4.2.3 Risk approach....65
4.2.4 Numerical variables....66
4.2.5 Numerical Micro Aggregation....67
4.2.6 Numerical Rank Swapping....68
4.3 Local suppression and making a safe microdata file....69
5 Description of Menu Items....71

5.1 Main window....71
5.2 The File Menu....72
5.2.1 File|OpenMicrodata....72
5.2.2 File|Exit....73
5.3 The Specify menu....73
5.3.1 Specify|MetaFile....73
5.3.2 Specify|Combinations....78
5.4 Modify....79
5.4.1 Modify|Show Tables....79
5.4.2 Global Recode....79
5.4.3 PRAM specification....82
5.4.4 Individual risk specification....83
5.4.5 Synthetic Data....84
5.4.6 Modify numerical variables....85
5.4.7 Numerical Micro Aggregation....86
5.4.8 Numerical Rank Swapping....87
5.5 Output....89
5.5.1 Output|Make Suppressed File....89
5.5.2 Output|View Report....90
5.6 Tools....91
5.6.1 Record Linkage....91
5.7 Help....92
5.7.1 Help|Contents....92
5.7.2 Help|News....92
5.7.3 Help|About....92
6 Further decriptions....94
6.1 Meta data file....94
6.2 Codelist file....95
6.3 Global recode file....96
7 Index....98

1 PREFACE

This is the users manual of version 4.3 of μ-ARGUS, a software package to assist in producing safe microdata. The package has been developed under Windows XP and runs under Windows versions from Windows 2000 and later. This version is the result of a continuous European cooperation effort starting in the fourth framework SDC-project and was continued in the fifth Framework CASC (Computational Aspects of Statistical Confidentiality) project. After that an update has been made as part of the CENEX-SDC project in 2006. This version is the result of the ESSNet-SDC project. Both the CENEX and the ESSNet project have been supported financially by Eurostat.

The purpose of the present manual is to give a potential user enough information so that he can understand the general principles on which μ-ARGUS is based, and also allow him to use the package. So it contains both general background information and detailed program information. For a more in-depth theoretical background we refer to the ESSNet SDC handbook.

μ–ARGUS is one part of a twin, the other one named τ–ARGUS. τ–ARGUS is a tool to help produce safe tables. Although the twins look quite different, from the inside they have a lot in common. In a sense they are Siamese twins, since their bodies have various organs in common. Nevertheless, a description of τ-ARGUS cannot be found in the present manual, but in a separate one.[1]

1.1 About the name ARGUS

Somewhat jokingly the name ARGUS can be interpreted as the acronym of 'Anti-Re-identification General Utility System'[2]. As a matter of fact, the name ARGUS was inspired by a myth of the ancient Greeks. In this myth Zeus has a girl friend named Io. Hera, Zeus' wife, did not approve of this relationship and turned Io into a cow. She let the monster Argus guard Io. Argus seemed to be particularly well qualified for this job, because it had a hundred eyes that could watch over Io. If it would fall asleep only two of its eyes were closed. That would leave plenty of eyes to watch Io. Zeus was eager to find a way to get Io back. He hired Hermes who could make Argus fall asleep by the enchanting music on his flute. When Hermes played his flute to Argus this indeed happened: all its eyes closed, one by one. When Hermes had succeeded in making Argus fall asleep, Argus was decapitated. Argus' eyes were planted onto a bird's tail - a type of bird that we now know under the name of peacock. That explains why a peacock has these eye-shaped marks on its tail. This also explains the picture on the cover of this manual. It is a

[1] Anco Hundepool et al 2012, *τ*–ARGUS user manual Version 3.5, Methodology Department, Statistics Netherlands, The Hague, The Netherlands.

[2] This interpretation is due to Peter Kooiman, and dates back to around 1992 when the first prototype of ARGUS was built by Wil de Jong.

copperplate engraving of Gerard de Lairesse (1641-1711) depicting the process where the eyes of Argus are being removed and placed on the peacock's tail.[3]

Like the mythological Argus, the software is supposed to guard something, in this case data. This is where the similarity between the myth and the package is supposed to end, as we believe that the package is a winner and not a looser as the mythological Argus is.

1.2 Contact

Feedback from users will help improve future versions of μ–ARGUS and is therefore greatly appreciated. Suggestions for improvements can be sent to Anco Hundepool (aj.hundepool@cbs.nl).

1.3 Acknowledgements

μ–ARGUS has been developed as part of the CASC project that was partly sponsored by the EU under contract number IST-2000-25069. This support is highly appreciated. The CASC (Computational Aspects of Statistical Confidentiality) project is part of the Fifth Framework of the European Union. Recent extensions of μ-ARGUS have been made possible during the European CENEX-SDC-projectError: Reference source not found (grant agreement 25200.2005.001-2005.619) and the ESSNet project. (grant agreement 25200.2005.003-207.670).

The main part of the μ–ARGUS software has been developed at Statistics Netherlands by Aad van de Wetering and Ramya Ramaswamy (who wrote the kernel) and Anco Hundepool (who wrote the interface and acted as the overall project leader). However this software would not have been possible without the contributions of several others, both partners in the CASC-project and later projects. These developments include the risk-approach, coordinated by Luisa Franconi from Istat and the Post Randomisation (PRAM) method (PRAM), based on work by Peter Kooiman, Peter-Paul de Wolf at CBS and Ardo van den Hout at Utrecht University. Further developments include work on Numerical Micro Aggregation and Rank Swapping by Josep Domingo and Vicenc Torra, and on synthetic data by Josep Domingo and.Krish Muralidhar

1.3.1 The CASC-project

The CASC project is the initiative in the 5th framework to explore new possibilities of Statistical Disclosure Control and to extend further the existing methods and tools. A key issue in this project is an emphasis more on practical tools, and the research needed to develop them. For this purpose a new consortium has been brought together. It has taken over the results and products emerging from the SDC-project. The main software developments in CASC are μ-ARGUS, the

[3] The original copy of this engraving is in the collection of 'Het Leidsch Prentenkabinet' in Leiden, The Netherlands.

software package for the disclosure control of microdata while τ-ARGUS handles tabular data.

The CASC-project will involve both research and software development. As far as research is concerned the project will concentrate on those areas that can be expected to result in practical solutions, which can then be built into (future version of) the software. Therefore the CASC-project has been designed round this software twin ARGUS. This will make the outcome of the research readily available for application in the daily practice of the statistical institutes.

CASC-partners

At first sight the CASC-project team had become rather large. However there is a clear structure in the project, defining which partners are working together for which tasks. Sometimes groups working closely together have been split into independent partners only for administrative reasons.

Institute	Short	Country
1. Statistics Netherlands	CBS	NL
2. Istituto Nationale di Statistica	ISTAT	I
3. University of Plymouth	UoP	UK
4. Office for National Statistics	ONS	UK
5. University of Southampton	SOTON	UK
6. The Victoria University of Manchester	UNIMAN	UK
7. Statistisches Bundesamt	StBA	D
8. University La Laguna	ULL	ES
9. Institut d'Estadistica de Catalunya	IDESCAT	ES
10. Institut National de Estadística	INE	ES
11. TU Ilmenau	TUIlm	D
12. Institut d'Investigació en Intelligència Artificial-CSIC	CIS	ES
13. Universitat Rovira i Virgili	URV	ES
14. Universitat Politècnica de Catalunya	UPC	ES

Although Statistics Netherlands is the main contractor, the management of this project is a joint responsibility of the steering committee. This steering committee constitutes of 5 partners, representing the 5 countries involved and also bearing a responsibility for a specific part of the CASC-project:

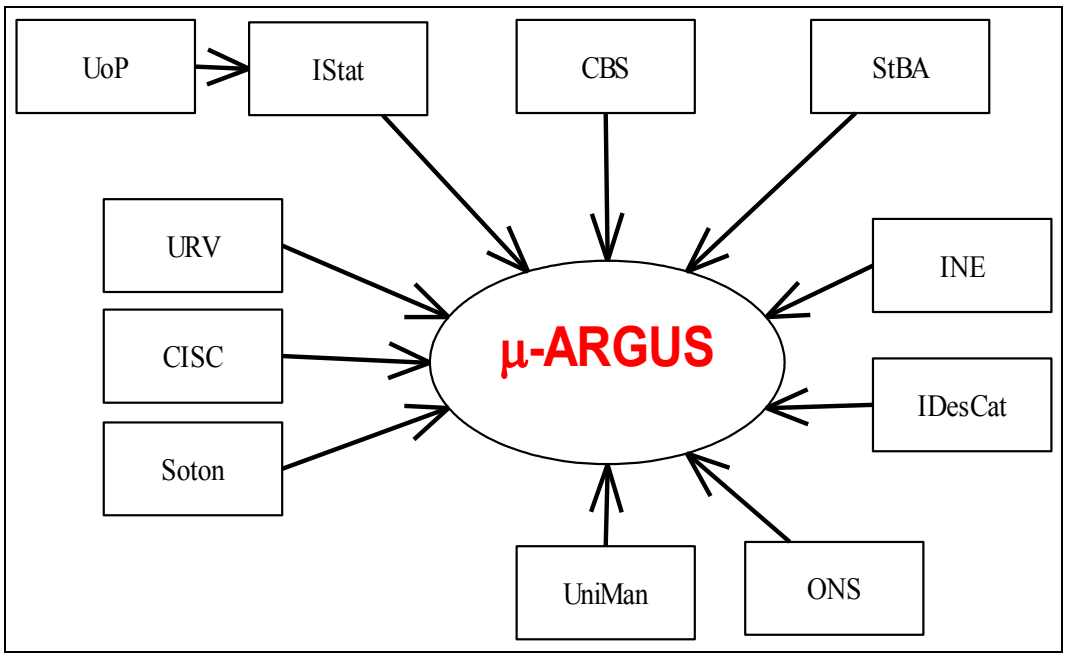
CASC Steering Committee

Institute	Country	Responsibility
Statistics Netherlands	Netherlands	Overall manager Software development
Istituto Nationale di Statistica	Italy	Testing
Office for National Statistics	UK	

Statistisches Bundesamt	Germany	Tabular data
Universitat Rovira i Virgili	Spain	Microdata

The CASC-microdata team

Several partners of the CASC-team work together and contribute in various stages to the development of μ-ARGUS. These contributions are either the actual software development, the research underlying this software development or the testing of the software.



1.3.2 The CENEX project

In 2006 thanks to the CENEX project[4] on SDC further extensions on the ARGUS software have been made. The CENEX (Centres of Excellence in Statistics) was an initiative by Eurostat to further foster the cooperation between the NSIs in Europe. The emphasis was more on using and promoting the currently available methods and tools, but also improving the tools by making them more easily usable in practical situations. Further testing of ARGUS has led to several improvements. Most strikingly the option of reading and writing directly a SPSS system file has now been included, but also many smaller improvements and corrections have been made.

The CENEX project was led by Statistics Netherlands. The NSIs of Italy, Germany, UK, Sweden, Slovenia, Estonia and Austria participated. The University Rovira i Virgili, Tarragona, also participated.

[4] See also the CENEX website: http://neon.vb.cbs.nl/casc/cenexindex.htm

1.3.3 The ESSNet project[5]

The ESSNet project, that can be seen as a continuation/follow-up of the CENEX project brought the development of μ-ARGUS another step forward. In the first release within the ESSNet we have corrected several errors and inconsistencies and also introduced the option of processing data from a SPSS system file. A step forward to enhance the usability of μ-ARGUS. Using μ-ARGUS in conjunction with SAS has been studies as well. Several reports are available on the ESSnet website

The testing of the members of the ESSNet team has contributed in several smaller improvements. The ESSNet project also made a first version of generating synthetic data possible and extended several methods with a blocking procedure allowing processing much larger data files. The synthetic data procedure is a new development in several ways, because this R-procedure is smoothly linked to μ-ARGUS.

[5] See also the ESSNet website at http://neon.vb.cbs.nl/casc/essnetindex.htm

2 INTRODUCTION

The growing demands from researchers, policy makers and others for more and more detailed statistical information leads to a conflict. The statistical offices collect large amounts of data for statistical purposes. The respondents are only willing to provide a statistical office with the required information if they can be certain that their data will be used with the greatest care, and in particular will not jeopardise their privacy. So statistical use of the data by outside users should not lead to a compromise of confidentiality. However, making sure that microdata cannot be misused for disclosure purposes, requires, generally speaking, that they should be less detailed, or modified in another way that hampers the disclosure risk. This is in direct conflict with the wish of researchers to have as detailed data as possible. Much detail allows not only more detailed statistical questions to be answered, but also more flexibility: the user can lump together categories in a way that suits his purposes best. The field of statistical disclosure control in fact feeds on this trade-off: How should a microdata set be modified in such a way that another one is obtained with acceptable disclosure risk, and with minimum information loss? And how exactly can one define disclosure risk? And how should one quantify information loss? And once these problems have been solved - no matter how provisionary - the question is how all this wisdom can actually be applied in case of real microdata. If a certain degree of sophistication is reached the conclusion is inescapable: specialised software is needed to cope with this problem and μ–ARGUS is such software.

In order to be able to use μ–ARGUS one should understand the ideas about disclosure risk, information loss, etc. on which it is based. Therefore the section 2 is devoted to giving the necessary background information. In later sections more detailed information about the working and use of μ–ARGUS can be found.

2.1 System Requirements

In general terms, the user requires a reasonably powerful Windows PC (Windows 2000 or later). Any libraries required will be installed automatically on installation, provided, of course, that the installer has appropriate installation rights, e.g. administrator rights on a Windows PC. This is a typically Windows feature. The installation of the new R-procedure for synthetic data however is different and described in the section on synthetic data (section 5.4.5).

3 PRODUCING SAFE MICRODATA

The purpose of the present chapter is to give the necessary background information for the use of μ–ARGUS. Producing safe micro data is not a trivial issue. It should first be explained when microdata are considered safe or unsafe. It should also be explained how unsafe data can be modified to become safe.

The Cenex-project made a start with composing a handbook on Statistical Disclosure Control. In the ESSNet project we continued to work on the handbook. This handbook will serve as a basic manual on the theory of SDC. Therefore it is recommended to use this handbook as a background reading. We will include references to the handbook were appropriate.

3.1 Safe and unsafe microdata

μ–ARGUS is based on a view of safety/unsafety of microdata that is used at Statistics Netherlands. In fact the incentive to build a package like μ–ARGUS was to allow data protectors at Statistics Netherlands to apply the general rules for various types of microdata easily, and to relieve them from the chore and tedium that producing a safe file in practice can involve. Not only should it be easy to produce safe microdata, it should also be possible to generate a logfile that documents the modifications of a microdata file.

When implementing μ–ARGUS it was the objective to produce a package that, on the one hand, is able to handle the specific rules that Statistics Netherlands applies to produce safe data, and on the other hand is more general. Nevertheless the generality has a bound. Despite unavoidable limitations, we believe that μ–ARGUS can justly be called a general-purpose package to help produce safe microdata. It is "general" within its philosophy, in the same sense as a Volkswagen Beetle is general in its capability of transporting a limited number of persons, but not 10 tons of bricks.

So we can say that the development of μ–ARGUS was heavily inspired by the type of rules that Statistics Netherlands uses to protect its microdata, rather than the precise form of the rules. So in order to be able to understand the general ideas that μ–ARGUS uses, it is necessary that the reader has a good understanding of the type of rules that it is supposed to support. A brief discussion of these is given below. A more extensive treatment can be found in Willenborg and De Waal (1996).[6]

The aim of statistical disclosure control is to limit the risk that sensitive information of individual respondents can be disclosed from data that are released to third party users. In case of a microdata set, i.e. a set of records containing information on individual respondents, such a disclosure of sensitive information of an individual respondent can occur after this respondent has been re-identified. That is, after it has been deduced which record corresponds to this particular individual. So, the aim of disclosure control should help to hamper re-identification of individual respondents represented in data to be published.

[6] Leon Willenborg and Ton de Waal (1996), Statistical Disclosure Control in Practice, Lecture Notes in Statistics Vol. 111, Springer-Verlag, New York. See in particular Section 4.2.9.

An important concept in the theory of re-identification is a key. A key is a combination of (potentially) identifying variables. An identifying variable, or an identifier, is one that may help an intruder re-identify an individual. Typically an identifying variable is one that describes a characteristic of a person that is observable, that is registered (identification numbers, etc.), or generally, that can be known to other persons. This, of course, is not very precise, and relies on one's personal judgement. But once a variable has been declared identifying, it is usually a fairly mechanical procedure to deal with it in μ–ARGUS.

In order to develop a theory of re-identification one should have a disclosure scenario.[7] Such a scenario is essentially a re-identification model, in which it is stated what sort of information an intruder is supposed to have, how he uses this information to re-identify individuals represented in a given microdata set, what sort of disclosures such a person is likely to make, and what motives he has for making such disclosures (he might want to know something about a particular individual, but his motive could also be to embarrass the data collector/releaser). More formally, such a disclosure scenario may lead to a disclosure risk model, i.e. a statistical model that gives a probability that disclosures are being made by an intruder, with an assumed level of knowledge of the population and for a given data file. Such disclosure risk models can range from a fairly simple, intuitive model to a rather sophisticated one. Of the latter type are models that either yield estimations of the number of unsafe combinations, or give for each record in the file a probability that an intruder will disclose it.

In a disclosure scenario, keys (as described above) are supposed to be used by an intruder to re-identify a respondent. Re-identification of a respondent can occur when this respondent is rare in the population with respect to a certain key value, i.e. a combination of values of identifying variables. Hence, rarity of respondents in the population with respect to certain key values should be avoided. When a respondent appears to be rare in the population with respect to a key value, then disclosure control measures should be taken to protect this respondent against re-identification. In practice, however, there is a problem. Often the datafile is only a sample of the population and rarity in the population (with respect to a certain key) in contrast to rarity in the data file is hard to establish. There is generally no way to determine with certainty whether a person who is rare in the data file (with respect to a certain key) is also rare in the population. Also an intruder may use another key than the key(s) considered by the data protector. For instance, the data protector may consider only keys consisting of at most three variables while the intruder may use a key consisting of four variables.

Therefore, it is better to avoid the occurrence of combinations of scores that are rare in the population instead of avoiding only population-uniques. To define what is meant by rare the data protector has to choose a threshold value D_k, for each key value k, where the index k indicates that the threshold value may depend on the key k under consideration. A combination of scores, i.e. a key value that occurs not more than D_k times in the population is considered unsafe, a key value that occurs more than D_k times in the population is considered safe. The unsafe combinations must be protected, while the safe ones may be published.

Re-identification of an individual can take place when several values of so-called identifying variables, such as ‘Place of residence’, ‘Sex’ and ‘Occupation’, are taken into consideration. The values of these identifying variables can be assumed to be known to relatives, friends, acquaintances and colleagues of a respondent. When several values of these identifying variables are combined a respondent may

[7] This concept is developed and used in Gerhard Paass and Udo Wauschkuhn, 1985, Datenzugang, Datenschutz und Anonymisierung, R. Oldenbourg Verlag, Munich.

be re-identified. Consider for example the following record obtained from an unknown respondent:

'Place of residence = Urk', 'Sex = Female' and 'Occupation = Statistician'.

Urk is a small fishing-village in the Netherlands, in which it is unlikely that many statisticians live, let alone female statisticians. So, when we find a statistician in Urk, a female statistician moreover, in the microdata set, then she is probably the only one. When this is indeed the case, anybody who happens to know this rare female statistician in Urk is able to disclose sensitive information from her record if such information is contained in this record.

There is a practical problem when applying the above rule that the occurrence (in the data file) of combinations of scores that are rare in the population should be avoided. Namely, it is usually not known how often a particular combination of scores occurs in the population. In many cases one only has the data file itself available to estimate the frequency of a combination of scores in the population. In practice one therefore uses the estimated frequency of a key value k to determine whether or not this key value is safe or not in the population. When the estimated frequency of a key value, i.e. a combination of scores, is greater than the threshold value D_k, then this combination is considered safe. When the estimated frequency of a key value is less or equal than the threshold value D_k, then this combination is considered unsafe. An example of such a key is 'Place of residence', 'Sex', 'Occupation'.

3.2 Statistical disclosure control measures

When a file is considered unsafe it has to be modified in order to produce a safe file. This modification can generally be done using so-called SDC techniques. Examples of such techniques are global recoding (grouping of categories), local suppression, PostRAndomisation Method (PRAM[8]), adding noise, microaggregation, top- and bottom coding. The effect of applying such techniques to a microdata set is that its information content is decreased. Variables are specified in fewer, and less detailed categories; values in certain records are suppressed; or, values in certain records are replaced by other values.

3.2.1 Global recoding[9]

In case of global recoding several categories of a variable are collapsed into a single one. In the above example we can recode the variable 'Occupation', by combining the categories 'Statistician' and 'Mathematician' into a single category 'Statistician or Mathematician'. When the number of female statisticians in Urk plus the number of female mathematicians in Urk is sufficiently high, then the combination 'Place of residence = Urk', 'Sex = Female' and 'Occupation = Statistician or Mathematician' is considered safe for release. Note that instead of recoding 'Occupation' one could also recode 'Place of residence' for instance.

[8] See P.-P. de Wolf, J.M. Gouweleeuw, P. Kooiman and L.C.R.J. Willenborg, Reflections on PRAM, Proceedings SDP98, Lisbon.

[9] See section 3.4.3.2 Global Recoding of the Handbook.

It is important to realise that global recoding is applied to the whole data set, not only to the unsafe part of the set. This is done to obtain a uniform categorisation of each variable. Suppose, for instance, that we recode the 'Occupation' in the above way. Suppose furthermore that both the combinations 'Place of residence = Amsterdam', 'Sex = Female' and 'Occupation = Statistician', and 'Place of residence = Amsterdam', 'Sex = Female' and 'Occupation = Mathematician' are considered safe. To obtain a uniform categorisation of 'Occupation' we would, however, not publish these combinations, but only the combination 'Place of residence = Amsterdam', 'Sex = Female' and 'Occupation = Statistician or Mathematician'.

3.2.2 Local suppression[10]

When local suppression is applied one or more values in an unsafe combination are suppressed, i.e. replaced by a missing value. For instance, in the above example we can protect the unsafe combination 'Place of residence = Urk', 'Sex = Female' and 'Occupation = Statistician' by suppressing the value of 'Occupation', assuming that the number of females in Urk is sufficiently high. The resulting combination is then given by 'Place of residence = Urk', 'Sex = Female' and 'Occupation = missing'. Note that instead of suppressing the value of 'Occupation' one could also suppress the value of another variable of the unsafe combination. For instance, when the number of female statisticians in the Netherlands is sufficiently high then one could suppress the value of 'Place of residence' instead of the value of 'Occupation' in the above example to protect the unsafe combination. A local suppression is only applied to a particular value. When, for instance, the value of 'Occupation' is suppressed in a particular record, then this does not imply that the value of 'Occupation' has to be suppressed in another record. The freedom that one has in selecting the values that are to be suppressed allows one to minimise the number of local suppressions.

3.2.3 Top and bottom coding[11]

Global recoding is a technique that can be applied to general categorical variables, i.e. without any requirement of the type. In case of ordinal categorical variables one can apply a particular global recoding technique namely top coding (for the larger values) or bottom coding (for the smaller values). When, for instance, top coding is applied to an ordinal variable, the top categories are lumped together to form a new category. Bottom coding is similar, except that it applies to the smallest values instead of the largest. In both cases, the problem is to calculate "tight" threshold values. In case of top coding this means that the threshold value should be chosen in such a way that the top category is as small as possible under the condition that this category is large enough to guarantee safety. In other words the threshold value should be chosen as large as possible, under the safety constraint. In case of bottom coding this threshold value should be as small as possible, of course. The safety is checked in terms of the frequencies of the value combinations that have to be checked.

Top and bottom coding can also be applied to continuous variables. What is important is that the values of such a variable can be linearly ordered. It is possible to calculate threshold values and lump all values larger than this value together (in

[10] See section 3.4.3.4 Local Suppression of the Handbook.
[11] See section 3.4.3.3 Top and bottom coding of the Handbook.

case of top coding) or all smaller values (in case of bottom coding). Checking whether the top (or bottom) category is large enough is also feasible.[12]

3.2.4 The Post RAndomisation Method (PRAM)[13]

General introduction

PRAM is a disclosure control technique that can be applied to categorical data. Basically, it is a form of deliberate misclassification, using a known probability mechanism. Applying PRAM means that for each record in a microdatafile, the score on one or more categorical variables is changed. This is done, independently of the other records, using a predetermined probability mechanism. Hence the original file is perturbed, so it will be difficult for an intruder to identify records (with certainty) as corresponding to certain individuals in the population. Since the probability mechanism that is used when applying PRAM is known, characteristics of the (latent) true data can still be estimated from the perturbed data file.

Another technique that is closely related to PRAM is the technique called Randomised Response (RR), see e.g. Warner (1965). Whereas RR is applied *before* the data are obtained, PRAM is applied *after* the data have been obtained. Both methods use known probability mechanisms to change scores on categorical variables. RR is used when interviewers have to deal with highly sensitive questions, on which the respondent is not likely to report true values in a face-to-face interview setting. By embedding the question in a pure chance experiment, the true value of the respondent is never revealed to the interviewer.

PRAM, the method

In this section a short theoretical description of PRAM is given. For a detailed description of the method, see e.g., Gouweleeuw et al. (1998a and 1998b). For a discussion of several issues concerning the method and its consequences, see e.g., De Wolf (1998).

Consider a categorical variable ξ in the original microdata file to which PRAM will be applied and denote the same categorical variable in the perturbed file by X. Moreover, assume that ξ (and hence X as well) has K categories, numbered 1, …, K. Define the transition probabilities involved in applying PRAM by $p_{kl} = \mathrm{P}(X = l \mid \xi = k)$, i.e., the probability that an original score k will be changed into the score l. PRAM is then fully described by the $K \times K$ matrix with entries p_{kl} with $k, l = 1, \ldots, K$. Applying PRAM then means that, given the score k in the original file in record r, the score will be replaced by a score drawn from the probability distribution $p_{k1}, \ldots, p_{kK}$. For each record in the original file, this procedure is performed independently of the other records.

Note that p_{kk} is the probability that the score k in the original file is left unchanged. Moreover, be aware of the fact that, since PRAM involves a chance experiment, different (perturbed) microdata files will be obtained when applying the same PRAM-matrix several times to a single original microdata file (each time starting with the unperturbed original microdata file).

Since the transition probabilities are known, unbiased estimates of contingency tables are easily obtained. Other more elaborate techniques will be needed to

[12] For the other values this is generally not feasible because there are too many. One should first categorize the values and apply top or bottom coding to this new, ordinal variable. Note that in practice there is no difference between a continuous variable and an ordinal variable with many categories, since we are dealing with finite populations.

[13] See section 3.4.6 PRAM of the Handbook.

correct for the PRAM-perturbation in more complex analyses as e.g., in loglinear models.

References on PRAM

Gouweleeuw, J.M., P. Kooiman, L.C.R.J. Willenborg and P.P. de Wolf (1998a), Post Randomisation for Statistical Disclosure Control: Theory and Implementation, Journal of Official Statistics, Vol. 14, 4, pp. 463 – 478.

Gouweleeuw, J.M., P. Kooiman, L.C.R.J. Willenborg and P.P. de Wolf (1998b), The post randomisation method for protecting micropdata, Qüestiió, Quaderns d'Estadística i Investigació Operativa, Vol. 22, 1, pp. 145 – 156.

Warner, S.L. (1965), Randomized Response; a survey technique for eliminating evasive answer bias, Journal of the American Statistical Association, Vol. 57, pp. 622 – 627.

De Wolf, P.P., J.M. Gouweleeuw, P. Kooiman and L.C.R.J. Willenborg (1998), Reflections on PRAM, Proceedings of the conference "Statistical Data Protection", March 25-27 1998, Lisbon, Portugal. This paper can also be found on the CASC-Website (http://neon.vb.cbs.nl/casc/ : Related Papers)

3.2.5 Numerical Micro Aggregation

Microaggregation is a family of statistical disclosure control techniques for *quantitative* (numeric) microdata, which belong to the substitution/perturbation category. The rationale behind microaggregation is that confidentiality rules in use allow publication of microdata sets if records correspond to groups of k or more individuals, where no individual dominates (*i.e.* contributes too much to) the group and k is a threshold value. Strict application of such confidentiality rules leads to replacing individual values with values computed on small aggregates (microaggregates) prior to publication. This is the basic principle of microaggregation.

To obtain microaggregates in a microdata set with n records, these are combined to form g groups of size at least k. For each variable, the average value over each group is computed and is used to replace each of the original averaged values. Groups are formed using a criterion of maximal similarity. Once the procedure has been completed, the resulting (modified) records can be published.

Consider a microdata set with p continuous variables and n records (*i.e.* the result of observing p variables on n individuals). A particular record can be viewed as an instance of $\boldsymbol{X}'=(X_1, \ldots, X_p)$ where the X_i are the variables. With these individuals, g groups are formed with n_i individuals in the i-th group ($n_i \geq k$ and $n = \sum_{i=1}^{g} n_i$). Denote by x_{ij} the j-th record in the i-th group; denote by $\overline{x_i}$ the average record over the i-th group, and by $\overline{x}$ the average record over the whole set of n individuals.

The optimal k-partition is defined to be the one that maximizes within-group homogeneity; the higher the within-group homogeneity, the lower the information loss, since microaggregation replaces values in a group by the group centroid. The sum of squares criterion is common to measure homogeneity in clustering. The within-groups sum of squares *SSE* is defined as

$$SSE = \sum_{i=1}^{g} \sum_{j=1}^{n_i} (x_{ij} - \overline{x_i})'(x_{ij} - \overline{x_i}).$$

The lower *SSE*, the higher the within-group homogeneity.

The total sum of squares is

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

In terms of sums of squares, the optimal *k*-partition is the one that minimizes *SSE*. A measure *L* of information loss standardized between 0 and 1 can be obtained from

$$L = \frac{SSE}{SST}$$

For a microdata set consisting of many variables, these can be microaggregated together or partitioned into several groups of variables. In the latter case, independent microaggregation for each group of variables is performed. Different results are obtained depending on whether and how variables are partitioned.

3.2.6 Multivariate fixed-size microaggregation

The method for multivariate fixed-size microaggregation implemented in μ-Argus tries to form homogeneous groups of records by taking into account the distances between records themselves and between records and the average of all records in the data set; this method will be called MDAV (multivariate microaggregation based on Maximum Distance to Average Vector). Given a group of variables, the MDAV algorithm is as follows:

Algorithm MDAV[14]

1. The average record $\bar{x}$ of all records in the data set is computed. The most distant record x_r to the average record $\bar{x}$ is considered (using the squared Euclidean distance).
2. The most distant record x_s from the record x_r considered in the previous step is found.
3. Two groups are formed around x_r and x_s, respectively. One group contains x_r and the *k-1* records closest to x_r. The other group contains x_s and the *k-1* records closest to x_s.
4. If there are at least *3k* records which do not belong to the two groups formed in Step 3, go to Step 1 taking as new data set the previous data set minus the groups formed in the previous instance of Step 3.
5. If there are between *3k-1* and *2k* records which do not belong to the two groups formed in Step 3, compute: a) the average record $\bar{x}$ of the remaining records; b) the most distant record x_r from $\bar{x}$; c) the group containing the *k-1* records closest to x_r and x_r; d) the other group containing the rest of records. Exit the Algorithm.
6. If there are less than *2k* records which do not belong to the groups formed in Step 3, form a new group with those records and exit the Algorithm.

[14] See section 3.4.5.4 The MDAV heuristic for multivariate microaggregation of the Handbook.

The above algorithm will be applied independently to each group of variables resulting from partitioning the set of variables in the data set.

If each variable in the data set is processed independently from the other variables (*i.e.* microaggregation is performed on a variable by variable basis, which results in univariate microaggregation), a polynomial shortest-path algorithm exists to find the exact solution to optimal univariate microaggregation (Hansen and Mukherjee, 2002). Due to its high memory and computing requirements, this optimal algorithm is only recommended for small data sets.

References on microaggregation

J. Domingo-Ferrer and J. M. Mateo-Sanz (2002) "Practical data-oriented microaggregation for statistical disclosure control", IEEE Transactions on Knowledge and Data Engineering, vol. 14, pp. 189-201.

J. Domingo-Ferrer and V. Torra (2001) "A quantitative comparison of disclosure control methods for microdata", in Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies (eds. P. Doyle, J. Lane, J. Theeuwes and L. Zayatz), Amsterdam: North-Holland, pp. 111-133.

S. L. Hansen and S. Mukherjee (2002) "A polynomial algorithm for univariate optimal microaggregation", IEEE Trans. on Knowledge and Data Engineering (to appear).

3.2.7 Numerical Rank Swapping

Although originally described only for ordinal variables (Greenberg, 1987), this method can be used for any numerical variable (Moore, 1996). First, values of variable V_i are ranked in ascending order, then each ranked value of V_i is swapped with another ranked value randomly chosen within a restricted range (*e.g.* the rank of two swapped values cannot differ by more than p% of the total number of records). This algorithm is independently used on each variable in the original data set.

Rank swapping has been identified as a particularly well-performing methods in terms of the tradeoff between disclosure risk and information loss (see the comparison Domingo-Ferrer and Torra (2001)).

References on rank swapping

J. Domingo-Ferrer and V. Torra (2001) "A quantitative comparison of disclosure control methods for microdata", in Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies (eds. P. Doyle, J. Lane, J. Theeuwes and L. Zayatz), Amsterdam: North-Holland, pp. 111-133.

B. Greenberg (1987) "Rank swapping for ordinal data", U. S. Bureau of the Census (unpublished manuscript).

R. Moore (1996) "Controlled data swapping techniques for masking public use microdata sets", U. S. Bureau of the Census (unpublished manuscript).

3.2.8 Synthetic Data[15]

Publication of synthetic —*i.e.* simulated— data was proposed long ago as a way to guard against statistical disclosure. The idea is to randomly generate data with the constraint that certain statistics or internal relationships of the original dataset should be preserved. We next review the two synthetic data generators included in μ-ARGUS.

[15] See section 3.4.7 Synthetic microdata of the Handbook.

3.2.8.1 Muralidhar-Sarathy hybrid generator

Hybrid data are a mixture of original data and synthetic data. Let V an original data set whose attributes are numerical and fall into confidential attributes $X\ (= X_1, \dots, X_L)$ and non-confidential attributes $Y\ (= Y_1, \dots, Y_M)$. Let V' be a hybrid data set obtained from V, whose attributes are $X'\ (= X'_1, \dots, X'_L)$ (hybrid versions of X) and Y.

Muralidhar and Sarathy (2008) proposed a procedure (called MS in the sequel) for generating hybrid data as follows

$$X'_j = \gamma + X_j\alpha^T + Y_j\beta^T + e_i, j = 1, \dots, n$$

MS can yield hybrid data preserving the means and covariances of original data. To that end, the following equalities must be satisfied:

$$\beta^T = \sum\nolimits_{YY}^{-1} \sum\nolimits_{YX} (I - \alpha^T)$$

$$\gamma = (I - \alpha)\bar{X} - \beta\bar{Y}$$

$$\sum\nolimits_{ee} = \left(\sum\nolimits_{XX} - \sum\nolimits_{XY} \sum\nolimits_{YY}^{-1} \sum\nolimits_{YX}\right) - \alpha\left(\sum\nolimits_{XX} - \sum\nolimits_{XY} \sum\nolimits_{YY}^{-1} \sum\nolimits_{YX}\right)\alpha^T$$

where I is the identity matrix and Σ_{ee} is the covariance matrix of the noise terms e.

Thus, α completely specifies the procedure. The authors of MS admit that α must be selected carefully to ensure that Σ_{ee} is positive semidefinite. They consider three options for specifying the α matrix:

1. Take α as a diagonal matrix with all values in the diagonal being equal. In this case, Σ_{ee} is positive semidefinite and the value of the hybrid attribute X'_i depends only on X_i, but not on X_j for $j \neq i$. All confidential attributes X_i are perturbed at the same level.
2. Take α as a diagonal matrix, with values in the diagonal being not all equal. In this case, X'_i still depends only on X_i, but not on X_j for $j \neq i$. The differences are that the confidential attributes are perturbed at different levels and there is no guarantee that Σ_{ee} is positive semidefinite, so it may be necessary to try several values of α until positive semidefiniteness is achieved.
3. Taking α as a non-diagonal matrix does not guarantee positive semidefiniteness either and the authors of MS do not see any advantage in it, although it would be the only way to have X'_i depend on several attributes among $(X_1, \dots, X_L)$. With $\mathbb{R}$-Microhybrid, the dependence of X'_i on the original confidential attributes is the one provided by the underlying IPSO method.

3.2.8.2 Microaggregation-based hybrid data

In (Domingo-Ferrer and González-Nicolás, 2009) an alternative procedure to generate hybrid data based on microaggregation was proposed. Let V be an original data set consisting of *n* records. On input an integer parameter $k \in \{1, \ldots, n\}$, the procedure described in this section generates a hybrid data set V'. The greater *k*, the more synthetic is V'. Extreme cases are: i) *k* = 1, which yields V' = *V* (the output data are exactly the original input data); and ii) *k* = *n*, which yields a completely synthetic output data set V'.

The procedure calls two algorithms:

- A generic synthetic data generator S(C, C', *parms*), that is, an algorithm which, given an original data (sub)set C, generates a synthetic data (sub)set C' preserving the statistics or parameters or models of C specified in *parms*.
- A microaggregation heuristic, which, on input a set or *n* records and parameter *k*, partitions the set of records into clusters containing between *k* and 2*k* − 1 records. Cluster creation attempts to maximize intra-cluster homogeneity.

Procedure 1 (Microhybrid(V,V', parms, k))

1. Call microaggregation(V, *k*). Let $C_1, \ldots, C_k$ for some *k* be the resulting clusters of records.
   1. For *i* = 1, ..., *k* call S(C_i, C'_i, *parms*).
   2. Output a hybrid dataset V' whose records are those in the clusters $C'_1, \ldots, C'_k$.
      1.

At Step 1 of procedure Microhybrid above, clusters containing between k and 2*k* −1 records are created. Then at Step 2, a synthetic version of each cluster is generated. At Step 3, the original records in each cluster are replaced by the records in the corresponding synthetic cluster (instead of replacing them with the average record of the cluster, as done in conventional microaggregation). The Microhybrid procedure can work with any synthetic data generator (and thus with any kind of data), but the μ-ARGUS implementation uses the IPSO generator and hence can only work with numerical data.

Role of parameter k

We justify here the role of parameter *k* in Microhybrid:

- If *k* = 1, and *parms* include preserving the mean of each attribute in the original clusters, the output is the same original data set, because the procedure creates *n* clusters (as many as the number of original records). With *k* = 1, even variable-size heuristics will yield all clusters of size 1, because the maximum intra-cluster similarity is obtained when clusters consist all of a single record.
- If *k* = *n*, the output is a single synthetic cluster: the procedure is equivalent to calling the synthetic data generator *S* once for the entire data set.
- For intermediate values of *k*, several clusters are obtained at Step 1, whose parameters *parms* are preserved by the synthetic clusters generated at Step 2. As *k* decreases, the number of clusters (whose parameters are preserved in the data output at Step 3) increases, which causes the output data to look more and more like the original data. Each cluster can be regarded as a constraint on the synthetic data generation: the more constraints, the less freedom there is for generating synthetic data, and the output resembles more the original data. This is why the output data can be called hybrid.

It must be noted here that, depending on the synthetic generator used, there may be a lower bound for k higher than 1. When using IPSO with $|X|$ confidential attributes and $|Y|$ non-confidential attributes, it turns out that k must be at least $2|X| + |Y| + 1$; otherwise there are not enough degrees of freedom for the generator to work.

References on synthetic data

Domingo-Ferrer, J., and González-Nicolás, Ú. (2009). Hybrid Microdata Using Microaggregation. Manuscript.

Muralidhar, K, and Sarathy, R, (2008). Generating sufficiency-based nonsynthetic perturbed data. Transactions on Data Privacy, 1(1):17–33, 2008. http://www.tdp.cat/issues/tdp.a005a08.pdf.

3.2.9 Record Linkage[16]

Empirical Disclosure Risk Assessment: Realistic Matching Scenarios

In the following we explain how to assess the disclosure risk of a data set using empirical matching techniques available in μ-ARGUS. The idea of this method is to assess the risk of disclosure of confidential information by a potential data intruder by carrying out realistic matching scenarios, as described below.

3.2.9.1 Mathematical modelling

In a database cross match, see Elliott and Dale (1999), the data intruder matches an external database A with the whole confidential database B. For this, he uses variables which the external data have in common with the confidential data, the so-called key variables.

To be a candidate for a possible assignment, it is necessary for a record pair $(a,b) \in A \times B$ that both records coincide in their values of some specified variables. In the following these variables are called blocking variables, since they divide the whole data into disjoint blocks. The aim of blocking data is on the one hand to reduce the complexity of the subsequent assignment procedure and the allocated main storage and on the other hand the number of mismatches. Detailed empirical investigations on blocking data in the context of disclosure control can be found in Lenz and Vorgrimler (2005). A blocking-tool is also available in μ-ARGUS

In a non-technical way, the concept of matching may be introduced as bringing together pairwise information from two records $a \in A$ and $b \in B$, taken from different data sources, that are believed to refer to the same individual. The records a and b are then said to be matched. In the following it is tried to minimise the number of mismatches. If we presume that a potential data intruder had knowledge about the participation of the searched units in the target survey, the problem of matching might be formulated in mathematical terms as follows: Find an injective mapping $\varphi : A \to B$, based on some distance measure $d : A \times B \to [0,1]$ (or alternatively based on some similarity measure $w : A \times B \to [0,1]$), which maps every record of A onto a near (or similar) record of B.

[16] See section 3.3.8 Record Linkage of the Handbook.

More precisely, the mapping can be defined by the following single objective assignment problem:

$$\text{Minimise} \quad \sum_{i=1}^{n}\sum_{j=1}^{m} d(a_i, b_j)x_{ij}, \qquad \textbf{(AP)}$$

$$\begin{aligned} \text{subject to} \quad & x_{ij} \in \{0,1\} \quad \text{for} \quad i = 1,\ldots,n; j = 1,\ldots,m, \\ & \sum_{j=1}^{m} x_{ij} = 1 \quad \text{for} \quad i = 1,\ldots,n \quad \text{and} \\ & \sum_{i=1}^{n} x_{ij} \leq 1 \quad \text{for} \quad j = 1,\ldots,m. \end{aligned}$$

The constraints of *(AP)* ensure that each record *a* of the external data *A* is one-to-one assigned to some record *b* of the target data *B*. That is, $x_{ij}=1$ if and only if a_i is connected with b_j. Therefore, it seems to be reasonable to assume that *A* possesses a smaller or equal number of records than *B*.

Once the coefficients $d(a_i,b_j)$ are calculated, we can solve the linear assignment problem using classical established methods such as the simplex method. For larger data blocks (typically generated when dealing with tax statistics) it is recommendable for reasons of efficiency that approximation heuristics should be used. Fortunately, the usage of appropriate heuristics yields results near the optimum solution of the assignment problem, see Lenz (2006).

References

Elliot, M., Dale, A. (1999). *Scenarios of attack: the data intruder's perspective on statistical disclosure risk, Netherlands Official Statistics, 6-10.*

Lenz, R., Vorgrimler, D. (2005). *Matching German turnover statistics*, *RDC working paper No. 4.* Wiesbaden.

Lenz, R. (2006). *Measuring the disclosure protection of micro aggregated business microdata - an analysis taking as an example the German Structure of Costs Survey. Journal of Official Statistics* ***22****, 681-710.*

Josep Domingo-Ferrer, Record linkage ,entry for the *Encyclopedia of Database Systems*, New York: Springer-Verlag, 2009, pp. 2353-2354.

3.3 The Risk approach

3.3.1 Introduction

When microdata files are released, it is possible that external users may attempt to breach confidentiality. For this reason it is necessary to apply some form of disclosure risk assessment and data protection. In social surveys, the observed variables are frequently categorical in nature, and often comprise public domain variables (such as sex, age, region of residence) that may allow identification. An intruder may use these variables that are referred to as *key variables,* to perform a *disclosure*. The definition of disclosure adopted in this framework uses the concept of *re-identification disclosure* (e.g. Chen and Keller-McNulty, 1998; Fienberg and

Makov, 1998; Skinner and Holmes, 1998; Duncan and Lambert, 1986; Willenborg and de Waal, 2001): by disclosure it is meant a *correct record re-identification* operation that is achieved by an intruder when comparing a target individual in a sample with an available list of units that contains individual identifiers such as name and address. Re-identification occurs when the unit in the released file and a unit in the register that an intruder has access to belong to the same individual in the population. The underlying hypothesis is that the intruder will always try to match a record in the sample and a unit in the register using the *key variables* only.

This document describes the *individual risk methodology* as introduced in an initial paper by Benedetti and Franconi (1998) and implemented in version 4.0 of μ-ARGUS. This approach defines a measure of disclosure risk per-record (namely, the *individual risk*) that can be used to protect selected records by *local suppression*. The *individual risk* of re-identification of unit *i* in the sample is defined as the probability of it being correctly re-identified, i.e. recognised as corresponding to a particular unit in the population. In social surveys, this risk of re-identification can be expressed through the concept of *unique* or *rare combinations* in the sample. A combination is a cell in the contingency table obtained by cross-tabulating the key variables. A key issue is to be able to distinguish between combinations that are at risk, for example sample uniques corresponding to rare combinations in the population, and combinations that are not at risk, for example sample uniques corresponding to combinations that are common in the population. Benedetti and Franconi (1998) propose a framework for definition and estimation of the individual risk using the sampling weights, as the usual instrument that national statistical institutes adopt to allow for inference from the sample to the population

Besides the one just mentioned, a number of proposals for defining and estimating a re-identification risk per record has been made in the last few years: Fienberg and Makov (1998), Skinner and Holmes (1998), Elamir and Skinner (2004) define, with different motivations, a log linear model for the estimation of the individual risk. Further discussion of the approach presented here is in Rinott (2003), Polettini (2003b), Franconi and Polettini (2004). A related approach is described in Carlson (2002), Elamir and Skinner (2004) and Polettini and Stander (2004).

After the risk has been estimated, protection takes place. To this aim, a threshold in terms of risk, e.g. probability of re-identification (see Section 3.3.3.2) is selected; units exceeding such a threshold are defined at risk, and μ-Argus applies local suppression to those individuals only, so as to lower their probability of being re-identified. Note that this approach allows to release the sensitive variables unchanged, while suppressing some of the key variables for some records.

The paper has two different components: a technical description of the individual risk methodology, and a guided tour through μ-Argus for the application of the individual risk methodology

The first part is devoted to the methodology. Section 3.3.2 introduces the definition of disclosure and the disclosure scenario. Section 3.3.3 is devoted to measures of risk for independent records: after introducing the basic notation (Section 3.3.3.1), the individual risk is presented in Sections 3.3.3.2 (definition), 3.3.3.3 (estimation) and 3.3.3.4 (approximation). Section 3.3.3.5 introduces a new concept of global risk of the whole file, namely the re-identification rate, that can be helpful in selecting a threshold for the individual risk; this topic is dealt with in Section 3.3.3.6. Section 3.3.4 discusses the release of household data: the household risk is presented in Section 3.3.4.1, while Section 3.3.4.2 indicates how to set the threshold for household data. Finally, Section 3.3.5 contains a discussion of some theoretical issues in the individual risk methodology.

The second part is devoted to application and guides the reader through the use of μ-Argus; in particular, Section 3.4 shows the main steps needed to produce a safe microdata file according to the individual risk methodology, that combines local suppression with the individual risk measure .

3.3.2 Disclosure and disclosure scenario[17]

As mentioned in the Introduction section, the definition of *disclosure* mimics the strategy of an intruder trying to establish a link between a unit in the sample *s* to be released and a unit in an available archive. Such an archive, or *register*, contains individual direct identifiers (*name, ID number, phone number…*) plus a set of variables called *identifying* or *key variables (sex, age, marital status…).* The intruder tries to match unit i in the sample with a unit i^* in the register by comparing their scores on the key variables. A *re-identification* occurs when, based on this comparison, a unit i^* in the register is selected as matching to *i and this link is correct*, e.g. i^* is the labelling of unit i in the population.

To define the *disclosure scenario*, the following assumptions are made. Most of them are conservative and contribute to the definition of a worst case scenario:

1. a *sample s* from a population P is to be released, and *sampling design weights* are available;
2. the archive available to the intruder covers *the whole population P*; consequently for each $i \in s$, the matching unit i^* does always exist in P;
3. the archive available to the intruder contains the individual direct identifiers and a set of categorical *key variables* that are also present in the sample;
4. the intruder tries to match a unit i in the sample with a unit i^* in the population register by comparing the values of the key variables in the two files;
5. the intruder has no extra information other than that contained in the register;
6. a *re-identification* occurs when a link between a sample unit i and a population unit i^* is established and i^* is actually the individual of the population from which the sampled unit i was derived; e.g. the match has to be a *correct match* before an identification takes place.

Moreover we add the following assumptions:

1 the intruder tries to match all the records in the sample with a record in the population register;

2 the key variables agree on correct matches, that is no errors, missing values or time-changes occur in recording the key variables in the two data archives.

3.3.3 Measures of risk

In this section we briefly introduce the basic concepts underlying the individual risk methodology within μ-Argus. For a detailed description of the individual risk methodology we refer to Benedetti and Franconi (1998), Benedetti, Franconi and Capobianchi (2003), Franconi and Polettini (2004).

[17] See section 3.3.2 Disclosure risk scenarios of the Handbook.

The concepts of individual and global risk are introduced, and technical details are provided.

3.3.3.1 Notation

Let the released file be a random sample s of size n selected from a finite population P consisting of N units. For a generic unit i in the population, we denote by $1/w_i$ its probability to be included in the sample.

Consider the contingency table built by cross-tabulating the key variables. A *combination* k is defined as the k-th cell in the contingency table. The set of combinations {1, …, k, …, K} defines a *partition* of both the population and the sample into cells. Observing the values of the key variables on individual $i \in s$ will classify such individual into one cell. We denote by $k(i)$ the index of the cell into which individual $i \in s$ is classified based on the values of the key variables. Typically, we expect to find several sampled units within the same combination k. We focus the analysis on each of the k=1, …, K cells of this contingency table.

Let f_k and F_k denote, respectively, the number of records in the released file and the number of units in the population with the k-th combination of categories of the key variables; F_k is unknown for each k. Depending on the key variables, the total number K of combinations can be quite high; in the sample to be released, only a subset of such number will be observed and only this subset of combinations for whom f_k>0, is of interest to the disclosure risk estimation problem.

3.3.3.2 Definition of disclosure risk for files of independent records

According to the concept of re-identification disclosure mentioned , we define the individual risk of disclosure of unit i in the sample as its *probability of re-identification*. In symbols:

$$\rho_i = \Pr\left(i \text{ correctly linked with } i^* \mid s, P\right). \qquad (1)$$

Clearly the probability that $i \in s$ is correctly linked with $i^* \in P$ is null if the intruder does not perform any link. Therefore we can condition on the event L_i : "the intruder attempts a re-identification of unit $i \in s$" and write

$$\rho_i = \Pr(i \text{ correctly linked with } i^* \mid s, P, L_i)\Pr(L_i),$$

where $\Pr(L_i)$ represents the probability that the intruder tries to establish a link between unit $i \in s$ and some unit in P.

The re-identification attempt is cast under the scenario described in Section 3.3.2. We adopt a pessimistic *scenario* by assuming that the intruder attempts to match all the records in the file ($\Pr(L_i)$=1 for all i); moreover we assume that the key variables agree on matching pairs, e.g. these are recorded without error or missing values in either s or P and moreover no time changes occur in the values of the key variables. The latter hypothesis raises the probability of re-identification of record i. Therefore the risk r_i that we get under this scenario is certainly not smaller than the risk ρ_i of formula (1):

$$\rho_i \le r_i = \Pr\left(i \text{ correctly linked with } i^* \mid s, P, \text{worst case scenario}\right) \quad (2)$$

Therefore measuring the disclosure risk by r_i in place *of* ρ_i is *prudential*, e.g. the actual risk is lower than the one we are estimating. We refer to r_i in (2) as the (base) *individual risk of re-identification*; this is the measure available in μ-Argus.

Recall that r_i is an upper bound to the probability of re-identification of unit i in the sample. Details on estimation of r_i are provided in the next section.

3.3.3.3 Estimation of the individual risk

We take into account the base individual risk of formula (2), that is, the upper bound to the probability of re-identification of a unit in the sample under the worst case scenario described in Section 3.3.2. As we already discussed, the risk is cast in terms of the cells of the contingency table built by cross-tabulating the key variables. Consequently all the records in the same cell have the same value of the risk; for this reason, for any record i in cell k we refer to r_k instead of r_i.

Looking at the k-th combination of the key variables, k=1, …, K, the intruder finds in the sample f_k individuals having such combination, out of F_k in the population. These individuals are exchangeable for re-identification, e.g. each of the F_k can be linked to any of the f_k. If we were to know the population frequency of the k-th combination, F_k, we would define the probability of re-identification simply by $1/F_k$. The agency that distributes the data may not have access to the population register, and may not know the population cell sizes F_k, therefore an inferential step is to be performed. We define the individual risk as *the agency estimate of the upper bound to the intruder's re-identification probability*. In the proposal by Benedetti and Franconi (1998), the uncertainty on F_k is accounted for in a Bayesian fashion by introducing the distribution of the population frequencies given the sample frequencies. The individual risk of disclosure is then measured as the (posterior) mean of $1/F_k$ with respect to the distribution of $F_k|f_k$:

$$r_i = \mathrm{E}\left(\frac{1}{F_k} \mid f_k\right) = \sum_{h \geq f_k} \frac{1}{h} \Pr(F_k = h \mid f_k)\,. \tag{3}$$

To determine the probability mass function of $F_k|f_k$, the following superpopulation approach is introduced (see Bethlehem *et al.*, 1990; Rinott, 2003; Polettini, 2003b):

$$\begin{aligned} &\pi_k : [\pi_k] \propto 1/\pi_k && \text{independently,} \quad k = 1, \cdots, \mathrm{K} \\ &F_k \mid \pi_k : \text{Poisson}(N\pi_k) && \text{independently,} \quad F_k = 0, 1, \cdots \\ &f_k \mid F_k : \text{binomial}(F_k, p_k) && \text{independently,} \quad f_k = 0, 1, \cdots, F_k. \end{aligned} \tag{4}$$

Under model (4), the posterior distribution of $F_k|f_k$ is negative binomial with success probability p_k and number of successes f_k. The law of $F_k|f_k$ is that of a negative binomial variable counting the number of trials before the j-th success, each with probability p_k. Its probability mass function is:

$$\Pr\left[F_k = h \mid f_k = j\right] = \binom{h-1}{j-1} p_k^j (1-p_k)^{h-j}, \quad h \geq j\,. \tag{5}$$

In Benedetti and Franconi (1998) it is shown that under the negative binomial distribution (5) the risk (3) can be expressed as

$$r_k = E(F_k^{-1} \mid f_k) = \int_0^\infty \left\{ \frac{p_k e^{-t}}{1 - q_k e^{-t}} \right\}^{f_k} dt \tag{6}$$

where $q_k = 1 - p_k$.

In the original formulation, the transformation $y = (1 - q_k e^{-t})^{-1}$ and the Binomial theorem were used in (6) to get

$$r_k = \left(\frac{p_k}{q_k}\right)^{f_k} \int_1^{1/p_k} \frac{1}{y}(y-1)^{f_k-1}\, dy = \left(\frac{p_k}{q_k}\right)^{f_k} \left\{ \sum_{j=0}^{f_k-2} (-1)^j \binom{f_k-1}{j} \frac{p_k^{\,j+1-f_k}-1}{f_k-j-1} + (-1)^{f_k} \log(p_k) \right\} \tag{7}$$

which is valid for f_k>1.

Alternatively, formula (6) can be expressed via the transformation y = exp(– *t*) as:

$$r_k = p_k^{\,f_k} \int_0^1 t^{f_k-1}(1-tq_k)^{-f_k}\, dt\,; \tag{8}$$

further, (8) can be rewritten (Polettini, 2003a) in terms of the hypergeometric function as

$$r_k = \frac{p_k^{\,f_k}}{f_k}\, {}_2F_1(f_k, f_k; f_k + 1;\, q_k) \tag{9}$$

where

$${}_2F_1(a,b;c;z) = \frac{\Gamma(c)}{\Gamma(b)\Gamma(c-b)} \int_0^1 t^{b-1}\left(1-t^{c-b-1}\right)\left(1-tz\right)^{-a} dt$$

is the integral representation (valid for $\Re(c) > \Re(b) > 0$) of the Gauss hypergeometric series (see Abramowitz and Stegun, 1965).

An estimate of the individual risk of disclosure (3) can be obtained by estimating p_k in (7) or (9). Given F_k, the maximum likelihood estimator of p_k under the binomial model in (4) is

$$\hat{p}_k = \frac{f_k}{F_k}\,.$$

F_k being not observable, Benedetti and Franconi (1998) propose to use

$$\hat{p}_k = \frac{f_k}{\sum_{i:k(i)=k} w_i}\,, \tag{10}$$

where $\sum_{i:k(i)=k} w_i$ is an estimate of F_k based on the sampling design, possibly calibrated (Deville and Särndal, 1992).

Generally speaking, use of formula (7) for the risk leads to numerically unstable estimates for values of $\hat{p}_k$ close to 0 or 1. Formula (9) does not suffer from this drawback. Using (10) in (9) we therefore end up with the following *plug-in estimate of the individual risk*:

$$\hat{r}_k = \frac{\hat{p}_k^{\,f_k}}{f_k}\, {}_2F_1(f_k; f_k; f_k + 1; 1 - \hat{p}_k)\,. \tag{11}$$

The negative binomial distribution is defined for 0 < pk < 1; in practice, the estimates $\hat{p}_k$ might attain the extremes of the unit interval. We never deal with $\hat{p}_k = 0$, as it corresponds to f_k = 0; on the other hand, if $\hat{p}_k = 1$, ${}_2F_1(f_k; f_k; f_k + 1; 1 - \hat{p}_k) = 1$, so that the individual risk equals $1/f_k$.

3.3.3.4 Approximation of the individual risk

For large values of the parameters f_k, 1- $\hat{p}_k$, numerical evaluation of the hypergeometric function can be computationally demanding. Polettini (2003b) has derived approximations that can be used even for moderate cell sizes. The approximations provided are based on the series representation of the hypergeometric function ${}_2F_1(f_k, f_k; f_k + 1; q_k)$. The former is divergent when f_k <0, therefore divergence is never of concern in practice. Absolute convergence of the series is guaranteed for $f_k > 1$. This constraint does never apply as we are dealing with an approximation; moreover the analytic expression for the risk when $f_k = 1$ is known and equals $-\log(p_k)\dfrac{p_k}{1-p_k}$.

For f_k =2 or 3 the analytic expressions of the risk are known as well:

f_k=2: $r_k = \dfrac{p_k}{q_k^{\,2}}\left(p_k \log p_k + q_k\right)$;

f_k=3: $r_k = \dfrac{p_k}{2q_k^{\,3}}\left(q_k(3q_k - 2) - 2p_k^{\,2}\log p_k\right)$.

The approximations to be proposed use the contiguity property of the hypergeometric function

$$ {}_2F_1(f_k, f_k; f_k+1; q_k) = (1-q_k)^{1-f_k}\,{}_2F_1(1,\,1;\,f_k+1; q_k) $$

(see Abramowitz and Stegun, 1965), and the series expansion:

$$ {}_2F_1(a,b;c;z) = \frac{\Gamma(c)}{\Gamma(a)\Gamma(b)}\sum_{n=0}^{\infty}\frac{\Gamma(a+n)\Gamma(b+n)}{\Gamma(c+n)}\frac{z^n}{n!}. $$

The formulae above lead to expressions of the type

$$ r_k = \frac{p_k}{f_k}\left(1 + \frac{q_k}{f_k+1} + \frac{2q_k^{\,2}}{(f_k+1)(f_k+2)} + \frac{6q_k^{\,3}}{(f_k+1)(f_k+2)(f_k+3)} + O(f_k^{-4})\right). \tag{12} $$

Approximations from below can be derived by truncating the series representation (12); the error depends on the remainder. Better accuracy may be achieved by introducing additional terms in the truncated series representation. In general, the approximation

$$ \frac{p_k}{f_k}\left(1 + \frac{q_k}{f_k+1} + \frac{2q_k^{\,2}}{(f_k+1)(f_k+2)}\right) \tag{13} $$

which has order O(f_k^{-3}), is accurate even for moderate cell sizes. Numerical assessment of the relative error $(r_k - r_k')/r_k$ has shown that the fourth order representation

$$ r_k' = \frac{p_k}{f_k}\left(1 + \frac{q_k}{f_k+1} + \frac{2q_k^{\,2}}{(f_k+1)(f_k+2)} + \cdots + \frac{7!q_k^{\,7}}{(f_k+1)(f_k+2)\cdots(f_k+7)}\right) \tag{14} $$

is satisfactory even for small cell sizes (see Fig. 1). Figure 1 also indicates that the truncated series approximation is not uniform in f_k and p_k and in general is more accurate for high values of f_k and small values of q_k.

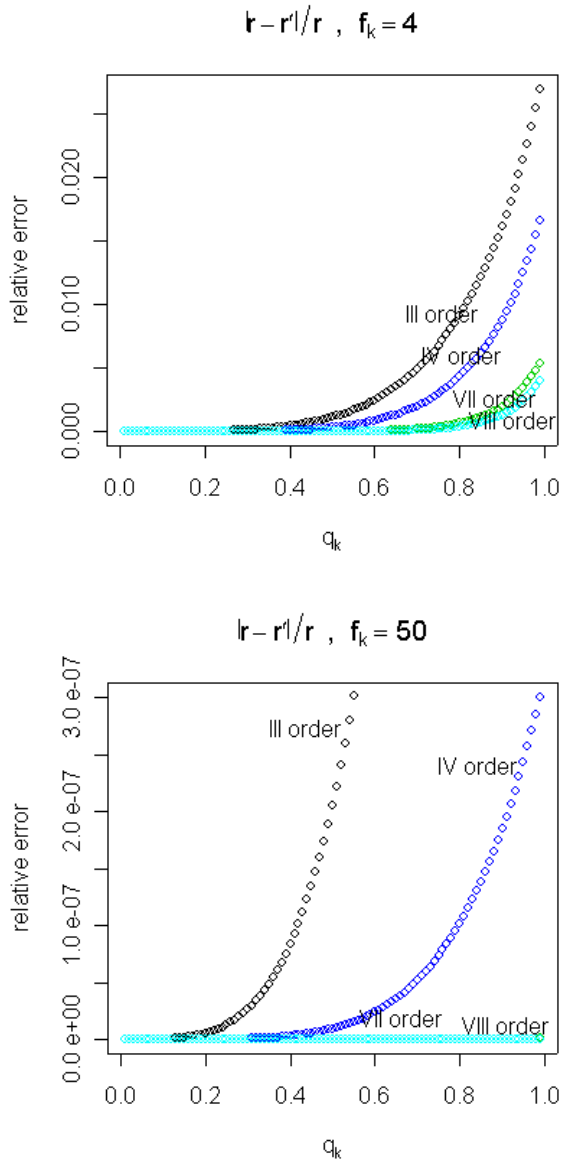


Figure 1. Relative errors of the approximation for different orders (a) $f_k = 4$ and (b) $f_k = 50$

Evaluation of the risk using the hypergeometric function requires specialised code and can prove computationally intensive for large values of the parameters; for these reasons and based on the previous findings, we suggest using formula (14) to evaluate the risk. Estimation of the risk can be performed by plug-in of the estimates $\hat{p}_k$ of formula (10) in the approximation (14).

3.3.3.5 Assessing the risk of the whole file: global risk

The individual risk provides a measure of risk *at the individual level*. A *global* measure of disclosure risk for the whole file can be expressed in terms of the expected number of re-identifications in the file (see Lambert, 1993). In this section we introduce the *expected number of re-identifications* and the *re-identification rate.* Whereas the first is a measure of disclosure that depends on the number of records in the file, the re-identification rate is independent of *n*.

Define a dichotomous random variable Φ, assuming value 1 if the re-identification is correct and 0 if the re-identification is not correct. In general for each unit in the sample one such variable Φ_i is defined, assuming value 1 with at most probability r_i. In the discussion we behave as if such probability was *exactly* r_i. The random variables Φ_i are not *i.i.d.*, but the risk is constant over the cells of the contingency table; therefore for combination k of the key variables we have f_k *i.i.d.* random variables Φ_i assuming value 1 when the re-identification is correct, with constant probability r_k. This probability can be exploited to derive the *expected number of re-identifications per cell*, which equals $f_k\ r_k$. In general, the overall expected number of re-identifications over the whole sample is

$$\mathrm{ER}=\sum_{i=1}^{n} E\left(\Phi_i\right)=\sum_{k=1}^{\mathrm{K}} f_k r_k \ .$$

The previous measure can be used to define the *re-identification rate* ξ as

$$\xi=\frac{1}{n}\mathrm{ER}=\frac{1}{n}\sum_{k=1}^{\mathrm{K}} f_k r_k \ . \qquad (15)$$

ξ provides a measure of *global risk, i.e.* a measure of disclosure risk for the whole file, that does not depend on the sample size and can be used to assess the risk of the file or to compare different types of release.

The *percentage of expected re-identifications*, i.e. the value ψ =100*ξ % provides an equivalent measure of global risk.

3.3.3.6 Application of local suppression within the individual risk methodology

μ-Argus contains a module that estimates the individual risk. As we have already seen, actually what is estimated is an upper bound for the probability of re-identification, thus leading to *prudential evaluation* of the risk. After the risk has been estimated, protection takes place. One option in protection is recoding the categories of selected variables (*global recoding*), the other is the application of *local suppression*. Local suppression consists of introducing missing values in some of the key variables of selected records; note that this choice allows to release the sensitive variables unchanged.

In μ-Argus the user can combine the technique of local suppression with either of two different measures of risk, both at the combination level: one is the sample frequency of the combination (used by the CBS methodology), the other is the risk of the individuals in that combination. In either event, protection by local suppression is applied to unsafe cells or combinations, and the user must input a *threshold* to classify these as either safe or unsafe. When applying local suppression in combination with the individual risk, users must select a threshold in terms of risk, e.g. probability of re-identification (see Section 3.3.3.2). Local suppression is applied to the unsafe individuals, so as to lower their probability of being re-identified. In order to select the risk threshold, that represents a level of *acceptable risk,* i.e. a risk value under which an individual can be considered safe, we suggest that the user refers to the *re-identification rate* defined in the previous section. Indeed the re-identification rate has a very natural interpretation, and a fixed range, whereas the range of the individual risk varies substantially with the data, and it is usually not possible to define a threshold that is valid for all data releases.

The user will therefore define a *release safe* when the expected rate of correct re-identifications is below a level he/she considers acceptable. As the re-identification rate is cast in terms of the individual risk (see formula (15)), a threshold on the re-

identification rate can be transformed into a threshold on the individual risk: this is described in Section 3.3.3.7. Under this approach, individuals are at risk because their probability of re-identification contributes a large proportion of expected re-identifications in the file.

Clearly, choice of the threshold affects the quality of the resulting "safe" file, and before saving the output file, the threshold can be changed. This will allow for assessment of the risk of the file, number of consequent suppressions and therefore quality of the "safe" file before data release. In order to reduce the number of suppressions, joint use of *global recoding* and local suppression is recommended. Recoding of selected variables will indeed lower the individual risks and therefore the re-identification rate of the file. The whole procedure of releasing a safe file indeed makes use of both global recoding and local suppression (see Section 3.4). The individual risk has here a twofold use: directly, it permits to identify unsafe records; indirectly, as r_i measures the contribution of record i to the re-identification rate ξ (or its equivalent ψ), it permits to assess whether the whole file can be safely released.

3.3.3.7 Threshold setting using the re-identification rate

In the previous section we remarked that a threshold for the individual risk can be determined by choosing the maximum tolerable *re-identification rate* in the sample. We next motivate this correspondence and show how this operation can be performed.

Consider the re-identification rate (15): cell k contributes to ξ an amount $r_k f_k$ of expected re-identifications. Since units belonging to the same cell k have the same individual risk, we can arrange cells in increasing order of risk r_k. We use the brackets to denote the generic element in this ordering; therefore the k-th element in this ordering will be denoted by the subscript (k).

Suppose that a *threshold* r^* has been set *on the individual risk*. Unsafe cells are those for which $r_k \geq r^*$. In the risk ordering of cells, we can find a cell index (K^*) (which is relative to r^*) that discriminates between safe and unsafe cells, i.e.

$$K^* \text{such that } r_{(K^*)} < r^* \text{ and } r_{(K^*+1)} \geq r^* \text{ ; (16)}$$

Formula (16) above puts r^* and K^* in one-to-one correspondence. Unsafe cells are indexed by $(k) = K^*+1, \ldots, K$, and local suppression ensures that after protection these cells will have risk below r^*. For the unsafe cells therefore $r_{(k)} < r^*$ *after protection.* Once protection is applied using threshold r^*, the expected number of re-identifications in the *released* sample is then certainly smaller than

$$\sum_{(k)=1}^{K^*} f_{(k)} r_{(k)} + r^* \sum_{(k)=K^*+1}^{K} f_{(k)} \ .$$

In fact, the threshold r^* can be picked from the discrete set of *observed* risk values, because choosing a threshold that is bracketed by two consecutive observed risk levels would not change the expected number of re-identifications.

The previous formula allows setting r^* so that the re-identification rate of the released file is bounded by a user-defined threshold value, i.e. $\xi < \xi^*$. A sufficient condition for this is that

$$\frac{1}{n}\left(\sum_{(k)=1}^{K^*} f_{(k)} r_{(k)} + r^* \sum_{(k)=K^*+1}^{K} f_{(k)} \right) < \xi^* . \quad (17)$$

Instead of selecting a threshold on the individual risk, it is more natural for the user to define a *threshold on the re-identification rate* ξ. Reversing the approach pursued so far, formula (17) can be exploited to determine a cell index K^* that keeps the re-identification rate ξ of the released file below τ. This in turn identifies a threshold on the individual risk: by relation (16), the corresponding individual risk threshold is $r_{(K^*+1)}$. The search of such a K^* is performed by a simple iterative algorithm.

3.3.4 Releasing files with a hierarchical structure

A relevant characteristic of social microdata is its inherent hierarchical structure, which allows us to recognise groups of individuals in the file, the most typical case being the *household*. Very often in social surveys the same information is collected for each household member and all this is stored in a single record that refers to the household. When defining the re-identification risk, it is important to take into account this dependence among units: indeed re-identification of an individual in the group may affect the probability of disclosure of all its members. So far, implementation of a hierarchical risk has been performed only with reference to households. We will therefore refer to individual and *household risk* for files of independent units and of households, respectively.

The individual risk methodology is currently the only approach that to some extent allows for a hierarchical structure in the data, which is in fact typical in many contexts. Under the hypothesis that the file has a hierarchical structure, it is possible to locate units within the household and, to a certain extent, also establish relationships between members of the same household. Allowing for dependence in estimating the risk enables us to attain a higher level of safety than when merely considering the case of independence. In the next section we introduce a measure that addresses this problem, in the effort to suit hierarchically structured, in particular household, microdata.

3.3.4.1 The household risk

For the household risk we adopt the same framework that we referred to in defining the individual risk. In particular, we use the concept of re-identification disclosure and the scenario described in Section 3.3.2. However, as external registers that contain information on households are not commonly available, for files of households we make the additional assumption that the intruder attempts a confidentiality breach by re-identification of *individuals* in households.

From the definition given in Section 3.3.3.2, it follows that the individual risk can be considered an estimate of a probability of re-identification. We define the *household risk* as the probability that *at least* one individual in the household is re-identified. By consequence, the household risk can be derived from the individual risks and knowledge of the household structure of the data. For a given household g of size $|g|$, whose members we label $i_1, \ldots, i_{|g|}$, we define the household risk as

$$r^h(g) = Pr(i_1 \cup i_2 \cup \ldots \cup i_{|g|} \text{ re-identified}) . \qquad (18)$$

Assuming independence of re-identification attempts within the same household, (18) can be expressed by Boole's formula using the individual risks $r_{i_1}, \ldots, r_{i_{|g|}}$ defined in (3):

$$r_g^{\ h} = \sum_{j=1}^{|g|} r_{i_j} - \sum_{i_j<i_l} r_{i_j} r_{i_l} + \sum_{i_j<i_l<i_m} r_{i_j} r_{i_l} r_{i_m} + \ldots + (-1)^{|g|+1} r_{i_1} r_{i_2} \ldots r_{i_{|g|}} \qquad (19)$$

By symmetry of the Boole's formula, the ordering of units in the group is not relevant. In a hierarchical file therefore the measure of disclosure risk is the same for all the individuals in household g and equals $r_g^{\ h}$. Estimation of the household risk can be performed by plug-in of the estimates of the individual risks $r_{i_1}, \ldots, r_{i_{|g|}}$ along the lines described in Section 3.3.3.

3.3.4.2 Household risk: threshold setting and identification of the unsafe records

Since all the individuals in a given household have the same household risk, the expected number of re-identified records in household g equals $|g| r_g^{\ h}$. We define the re-identification rate in a hierarchical file as

$$\xi^h = \frac{1}{n} \sum_{g=1}^{G} |\, g \,|\, r_g^{\ h} \ ,$$

where G is the total number of households in the file. The re-identification rate can now be used to define a threshold r^{h*} on the household risk r^h, much in the same way as in Section 3.3.3.7.

Note that the household risk $r_g^{\ h}$ of household g is computed by the individual risks of its household members. For a given household, it might happen that a household is unsafe ($r_g^{\ h}$ exceeds the threshold) because just one of its members, i, say, has a high value r_i of the individual risk. In order to protect the households, our approach is therefore to protect individuals in households, first protecting those individuals who contribute most to the household risk. For this reason, inside *unsafe households*, we need to identify *unsafe individuals*. In other words, we need a way to transform a threshold on the household risk r^h into a threshold on the individual risk r. To this aim, we notice that by formula (19) the household risk is bounded by the sum of the individual risks of the members of the household:

$$r_g^{\ h} \le \sum_{j=1}^{|g|} r_{i_j} \ .$$

Consider to apply a threshold r^{h*} on the household risk. In order for household g to be classified safe (i.e. $r_g^{\ h} < r^{h*}$) it is *sufficient* that all of its components have individual risk less than

$$\delta_g = r^{h*}/|g|.$$

This is clearly a strongly prudential approach, as we check whether a *bound* on the household risk is below a given threshold.

It is important to remark that the threshold δ_g just defined depends on the size of the household to which individual i belongs. This implies that for two individuals that are classified in the same combination k of key variables (and therefore have the same individual risk r_k), but belong to different households with different sizes, it might happen that one is classified safe, while the other unsafe.

In practice, denoting by $g(i)$ the household to which record i belongs, the approach pursued so far consists in turning a threshold r^{h*} on the household risk into a *vector of thresholds* on the *individual risks* r_i; i = *1*, …, *n*:

$$\delta_g = \delta_{g(i)} = r^{h*}/|g(\ i\ )|.$$

Individuals are finally set to unsafe whenever $r_i \geq \delta_{g(i)}$; local suppression is then applied to those records, if requested. Suppression of these records ensures that after protection the household risk is below the threshold δ_g.

3.3.5 Some Comments on Risk Estimation

The procedure relies on the assumption that the available data are a sample from a larger population. The sampling design is assumed to be known, as far as the sampling weights are concerned at least. If the sampling weights are not available, or if data represent the whole population, the strategy used to estimate the individual risk is not meaningful. Therefore we recommend using the individual risk methodology only for sample data, when sampling weights are available.

Recall that the household risk $r_g^{\ h}$ of household g only depends on the risks $r_{i_1},\ldots,r_{i_{|g|}}$ of its components, and these are evaluated under the hypothesis of no hierarchical structure in the file. For household data it is therefore important to include in the key variables that are used to estimate the risks $r_{i_1},\ldots,r_{i_{|g|}}$ also the available information on the household, such as the number of components or the household type. Note that when the household size is used as one of the key variables, a single threshold δ_g for each cell is defined, as all the individuals in the same cell have the same household size. Under this circumstance records in cell k are all safe or all unsafe (contrast with the remark of Section 3.3.4.2).

Suppose one computes the risk using the household size as the only key variable in a household data file, and that such file contains households whose risk is above a fixed threshold. Since information on the number of components in the household cannot be removed from a file with household structure, these records cannot be safely released, and no suppression can make them safe. This permits to check for presence of very peculiar households (usually, the very large ones) that can be easily recognised in the population just by their size and whose main characteristic, namely their size, can be immediately computed from the file.

In Di Consiglio et al. (2003) an experiment was conducted to assess the performance of the individual risk of disclosure. The aim was to investigate whether the individual risk is estimating the correct quantity, i.e. the real risk of an individual, and also whether the quality of this estimation is appropriate. A good agreement was noticed between the real risk and its estimates, although the precision of the estimator seems poor for rare combinations in the sample as compared to the more common ones. A minor precision is an intrinsic problem of small counts. Discriminating rare and common features in the population is, by far, the most difficult task especially when one can count on only one occurrence in the sample. For this reason further studies to improve the performance of the estimator used for the individual risk have been planned. Model (4) was questioned to provide a good fit to real data under some circumstances (e.g. Rinott, 2003). Our experiments seem to indicate that the model holds, at least with the data we use. Notice that the assumed model is compatible with a large number of relatively small cells in the population. These might occur with both small sized population and large number of combinations of key variables.

3.4 A guided tour of μ-Argus for the application of the Individual Risk Methodology

A general guided tour of μ-ARGUS can be found in Chapter 4. The special character of the risk-model justifies a special introduction here. This Section describes application of the individual risk methodology within μ-Argus version 4.0. Basically, the method detailed in the previous sections defines a measure of re-identification risk per record that is estimated using the sampling design weights. We distinguish here two different types of application: the first concerns files of *independent records*, whereas the second concerns files with a *household structure*. The latter must have a special structure that we discuss in Section 3.4.1. A household structure is detected by the software through the presence of a **household identifier** variable, i.e. a counter that distinguishes different households. This information has to be imputed in the metadata window (see Section 3.4.1). A household file usually also presents **household variables**, i.e. variables that take the same level for any member of the household.

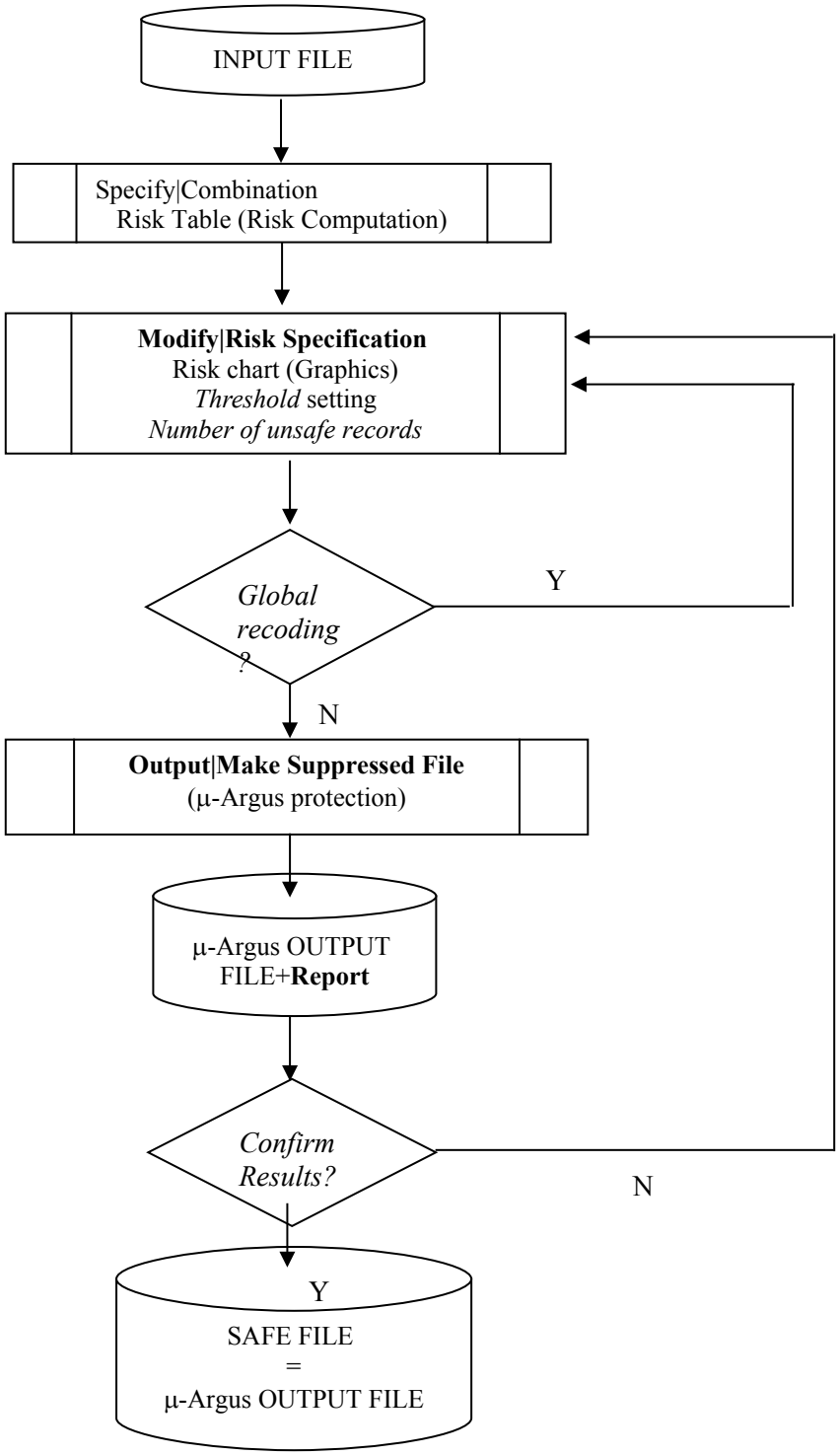
For files of independent records, μ-Argus computes the individual risk measure described in Section 3.3.3; for household data, μ-Argus automatically computes a household risk as described in Section 3.3.4.

Once the proper risk measure has been estimated, *local suppression* is applied to the individuals whose risk exceeds a given threshold. This is to be set by the user (see Sections 3.3.3.6 and 3.3.4). Local suppression lowers the individual risk of the protected records, and therefore the global risk of the file, as measured by the re-identification rate. To reduce the number of suppressions, *global recoding* can also be applied. Global recoding affects all the records in the file, thereby reducing their risks; as a consequence, the re-identification rate or global risk of the file is decreased.

We next discuss the steps needed to perform the protection of a data file under the individual risk model. Generally speaking, μ-Argus requires the following elements to be specified:

- the data and its structure;
- the *key variables*;
- a *risk table* based on the observed combinations of the key variables, that contains the individual risks. As the individual risk is the same for all the individuals having the same combination of key variables, the risk table can be built on combinations;
- a *threshold* that permits to classify records into safe/unsafe, respectively.

Recall that the *key variables* are public domain variables, such as sex, age, region of residence, that may allow identification; we assume that these are available to the intruder when attempting a disclosure.



The next Sections discuss the main ingredients of the procedure, presented following the steps that a user of μ-Argus has to take in order to protect the data

using the individual risk method together with local suppression and, possibly, global recoding. The graph shows the main lines along which the individual risk methodology proceeds.

3.4.1 Specification of a household structure in the Menu Specify|Metadata

The procedure for importing files of independent records does not differ from what is described in Section 4 of the user manual of μ-Argus. The menu **Specify|Metadata** is indeed common to any protection procedure applied by the software. For this reason we only discuss the case when the file to be protected has a *household structure*.

First of all, the data file must be presented in the form of a *file of individuals*, in which an additional variable, namely the **household identifier**, contains the household structure of the data. For an example of one such file, see the scheme in Figure 2. A few lines of the SAMHH.ASC demonstration file (that can be found in the hhdata directory of the μ-Argus installation directory) are also shown in Figure 3 to see how the input file looks like. The household identifier (highlighted in both figures) is simply a counter that permits to identify the household and to distinguish different households.

hhident	region	sex	marstat	age	hhtype
1	1	2	1	43	2
2	7	1	3	40	2
3	3	1	1	47	5
3	3	2	1	48	5
4	5	2	1	63	11
4	5	1	1	61	11
4	5	1	1	57	11
4	5	1	1	48	11
4	5	2	1	37	11
5	5	1	2	61	8
5	5	2	2	64	8
5	5	2	1	32	8
5	5	2	1	30	8
6	8	2	6	57	10
6	8	2	1	32	10
6	8	1	4	38	2
7	11	1	6	77	3

Figure 2. Structure of a household data file. **hhident** is the household identifier whereas **region** and **hhtype** represent the household variables "region of residence" and "household type".

```
122.2221 11152   0 40
122.2221 32401 150 40
122.2221 12112   0 40
266.6662 22762   0 76
266.6662 21712   0 76
355.5552 213911588 39
355.5552 22402   0 39
355.5552 11132   0 39
355.5552 12112   0 39
466.6662 11221 878 22
511.1111 21261 493 26
511.1111 22271 168 26
511.1111 11 22   0 26
511.1111 12 22   0 26
511.1111 12 12   0 26
677.7772 213611477 36
677.7772 22362   0 36
677.7772 12 92   0 36
677.7772 12 82   0 36
744.4442 114112729 41
855.5552 21241 979 24
855.5552 22222   0 24
855.5552 12 02   0 24
```

Figure 3 Structure of the **hhdata**. Highlighted is the household identifier variable.

It is important to remark that the input file must be arranged into households, in the sense that it is necessary that at least the members of the same household are grouped together in the input file. A sufficient condition for that is to sort the file by the value of the household identifier variable. If this is not the case, the household risk that Argus computes will be unreliable.

The information on the household identifier, which is crucial to make Argus aware that a household structure is present in the data, has to be entered in the **Specify| Metadata** window (setting for this variable the type **HH identifier**, see Figure 4). A household data file generally also presents **household variables** (variables that take the same level for all the members of the household, see Figure 2 and Section 2.7 of the manual); this information can be specified in the same window. The other steps do not differ from the usual data import procedure.

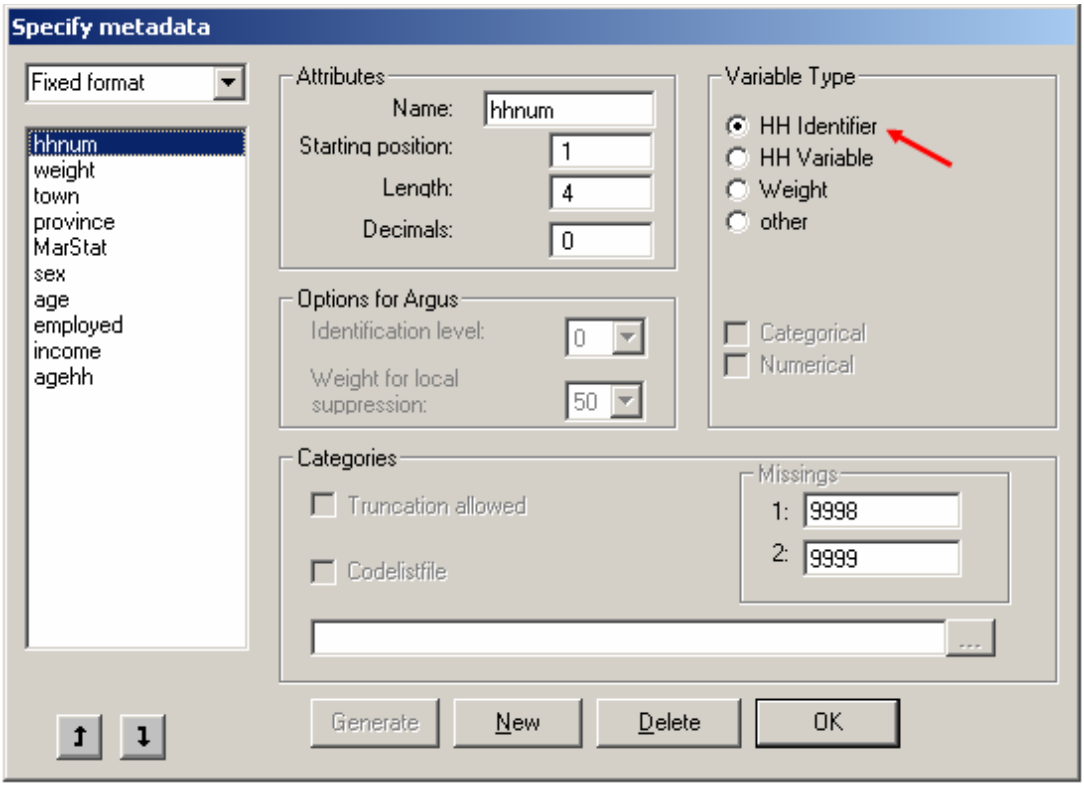


Figure 4. Specifying the household identifier variable **hhnum** to let μ-Argus read the household structure of the file

3.4.2 Menu Specify|Combinations

From the menu **Specify|Combinations**, users access a multi-purpose window (**Select Combinations**, see Figure 5) where the key variables can be selected. These represent the information that is assumed to be available to the intruder and therefore depend on the scenario. The software completes this step by building the contingency table obtained by cross-classifying the selected key variables; if the user requires to apply the *individual risk methodology*, μ-Argus additionally stores a so-called **risk table**, i.e. a contingency table in which a risk value is associated to each cell or combination. Note that μ-Argus allows specifying separate subgroups of key variables; in this circumstance, the software computes as many contingency tables as there are groups of key variables. When the traditional (Dutch) method is chosen, μ-Argus computes the unsafe combinations according to the concept of *uniques* or *fingerprints* as described e.g. in Willenborg and de Waal (1996; 2001). Under the individual risk approach, the unsafe records are those exceeding a given risk threshold; this can only be specified after the risk has been computed (see Section 3.4.3)

From the left panel of the window, the key variables can be moved to the central box using the button **1** in Figure 5. Move further the selected set of key variables to the right panel, using the arrow button indicated by **2** and press the button **Set table for risk model** (**3**). An **R** appears in the right panel of the window (indicated by a circle in Figure 5), indicating that the individual risk methodology will be applied. With this option μ-Argus is ready to build the above mentioned risk table. Recall that if the **R** option is not activated in the right panel, the traditional (Dutch) approach to identify unsafe records or combinations is pursued.

Suppose a user wants to apply the individual risk methodology using a single set of say, 4 key variables, e.g. REGION*SEX*AGE*MARSTAT (182*2*60*4 classes, see Figure 5). When the **Select Combinations** procedure is completed, μ-Argus computes the sample and population frequencies of the combinations of all 4 key variables, and attach a risk value to each combination.

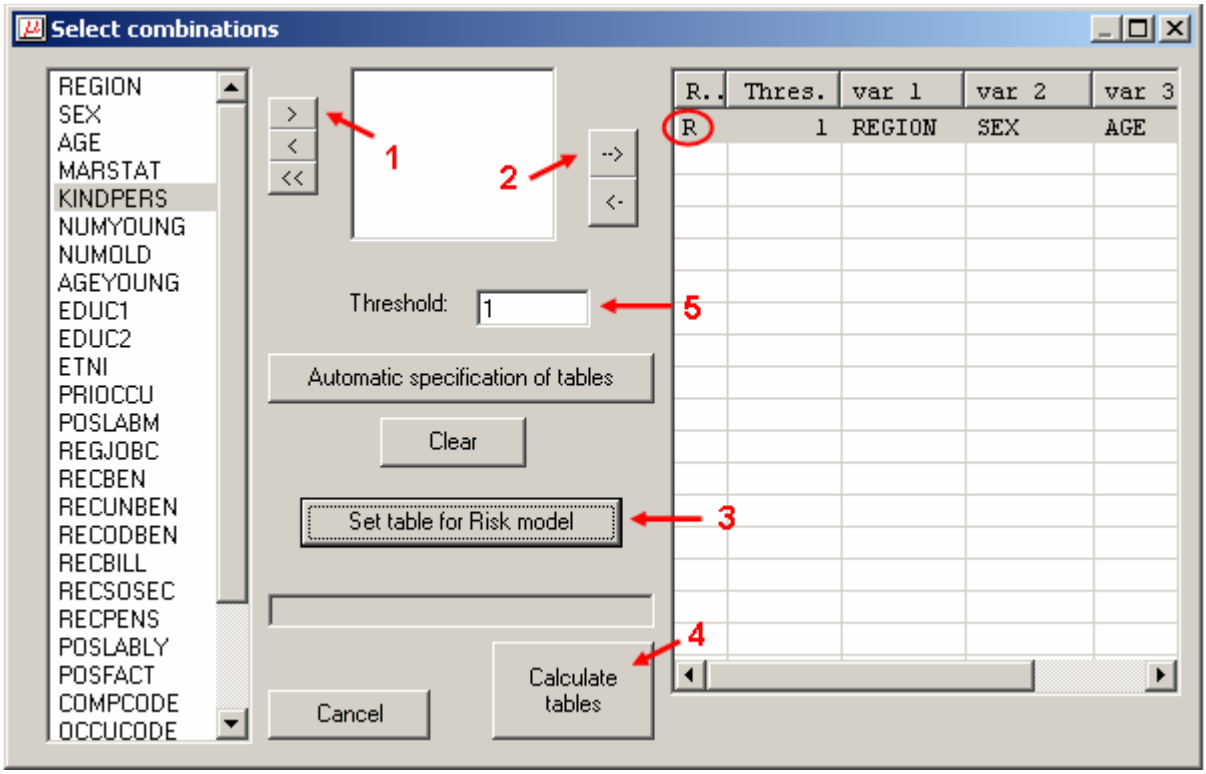


Figure 5. Select combinations window

Although this is not the recommended option, the software allows users to specify more than a single table, unless these overlap. Figure 6 below shows an instance of specifying two separate tables; recall that these must not have key variables in common. Suppose that a user has specified two non-overlapping sets of two key variables, e.g. REGION*SEX plus AGE*MARSTAT; the combinations to be considered in the two risk tables pertain to two separate two-way contingency tables, consequently 4-fold combinations are not taken into account. The user must decide in advance whether his/her model for the intruder accounts for knowledge of the set of all four key variables, or if the hypothesis is made that the intruder knows either set of two key variables. This latter instance is also equivalent to protecting data against two alternative attacks. In this case the procedure is split, and the steps to be described next are doubled. Of course, specifying the set of all four key variables is conservative i.e. safer, as more information is assumed to be available to the intruder (this is the recommended approach).

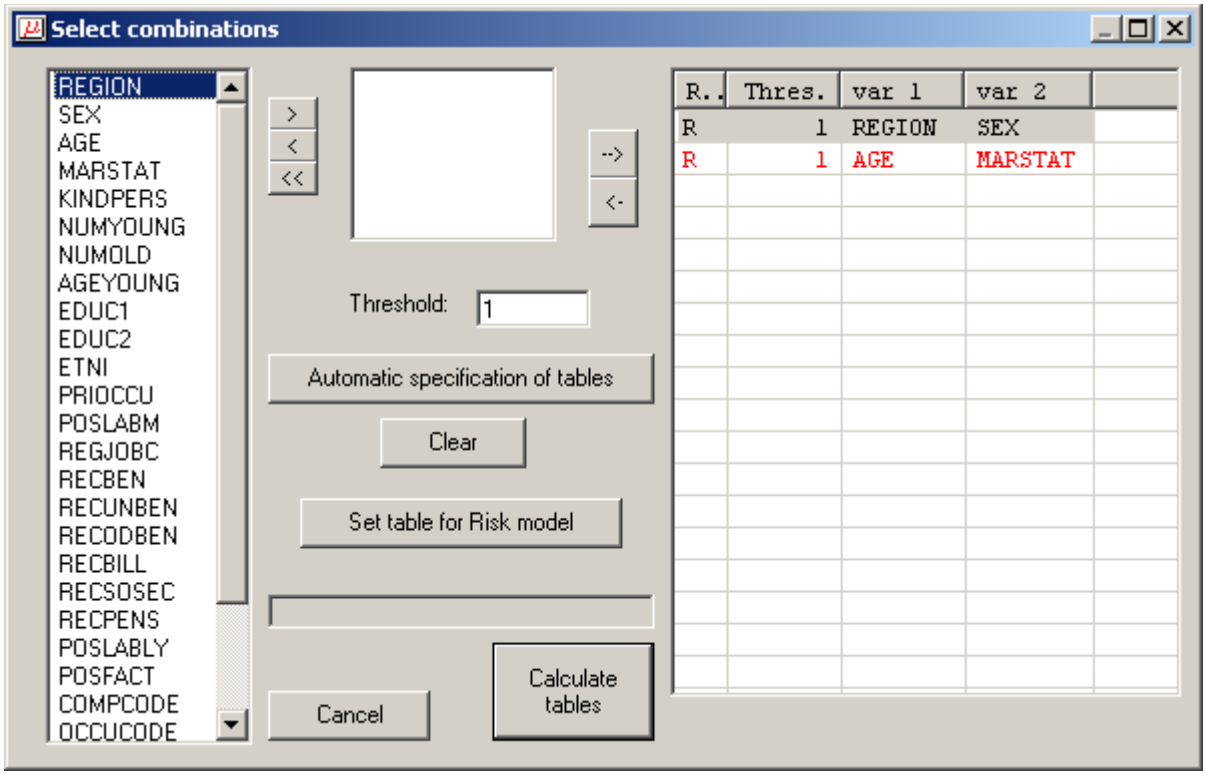
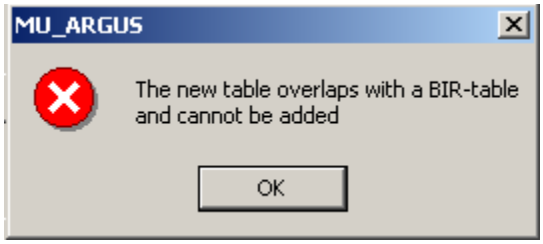


Figure 6. Selection of two sets of key variables

If overlapping risk tables are defined, i.e. if the user specifies two or more sets of key variables with at least one common variable, μ-Argus returns an alert message such as the one shown here. After that, either the introduction of an additional risk table is inhibited, or μ-Argus forces the user to remove the overlapping table(s). Recall also that when multiple risk tables are defined, the button **Set table for risk model** has to be used for every single table.



To start the procedure that computes the risk table and close the **Select Combinations** window, press the button **Calculate tables** (indicated by number **4** in Figure 5) . The software runs the routine that computes the risks and shows an **output window**.

3.4.2.1 Output window

Being common to any procedure run in μ-Argus, this window (see for instance Figure 7) is not directly connected with the risk methodology. Indeed it reports the number of unsafe combinations computed according to the *traditional (Dutch) methodology* which relies on the concept of uniques or fingerprints. Recall that in the traditional framework the number of unsafe records depends on the *frequency threshold*: a combination is unsafe if its frequency is less or equal than a given threshold value, that is imputed in the **threshold** box of the **Select Combination**

window (see Figure 5, number **5**). Note that the settings of the **Select Combinations** window would apparently allow to adopt *both* the traditional (Dutch) approach based on the frequency threshold *and* the individual risk approach. Indeed even when the individual risk methodology is adopted, the frequency **threshold** box (see Figure 5, number **5**) is active. However in this case the software does not make use of the selected threshold value, as it has no role in risk computation: any value of the frequency threshold has the same effect on risk computation, although the standard output given by μ-Argus (shown in Figure 7) changes according to the frequency threshold.

The output also differs according to whether one or more risk tables have been specified: in either case the left panel shows the **number of unsafe records** (defined according to the traditional methodology) for each **dimension**, e.g. level of cross-tabulation of the key variables.

As already explained, the default output does not return information specific to the risk model, nonetheless it can be useful for a generic assessment of the file. For instance, screening the left panel can suggest which variables may undergo recoding, whereas the right panel may help identifying the categories to be collapsed.

In our example, when all four key variables are selected into one single risk table, the output contains the number of unsafe records for each dimension up to 4 (Figure 7).

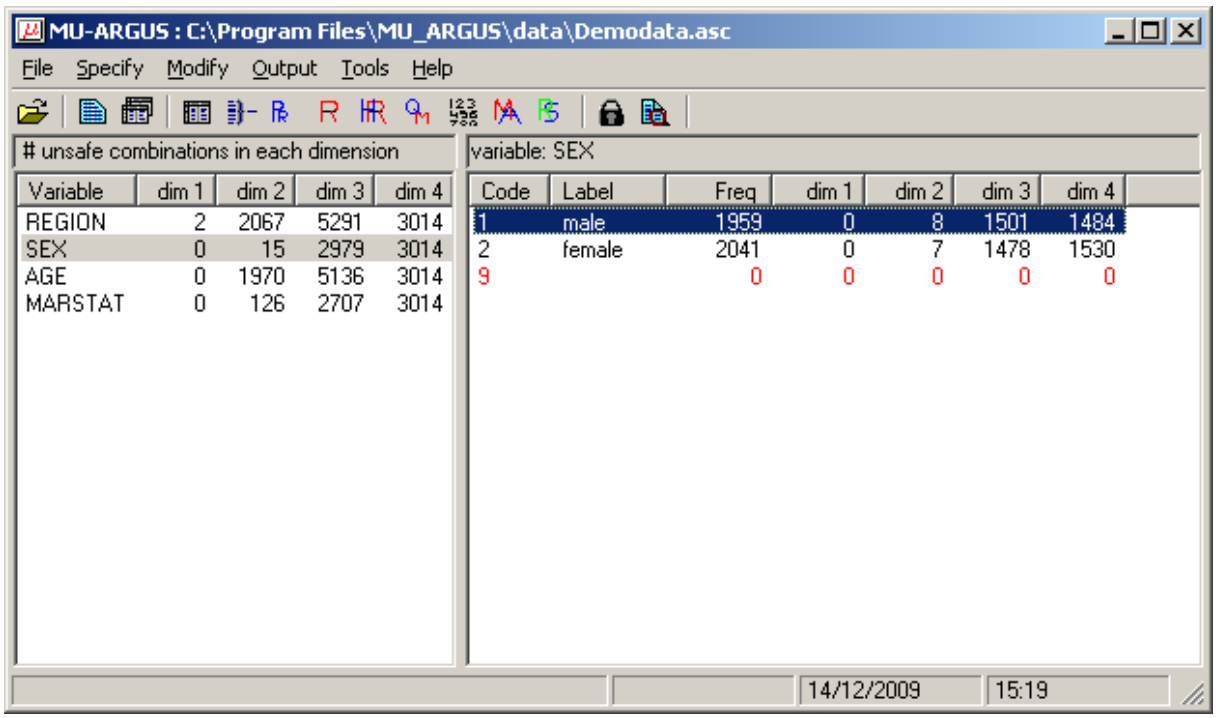


Figure 7. Output window for the individual risk methodology (4 key variables)

When we define two separate risk tables, each having two key variables, a single output window (see Figure 8) shows both tables. Compare the output shown in Figure 7: although the key variables are the same in both windows, in Figure 8 combinations up to dimension 2 are shown. In the latter case the two tables are indeed considered separately. Clearly the highest dimension considered in the

output window depends on the maximum level of cross-classification allowed in each of the risk tables defined by the user.

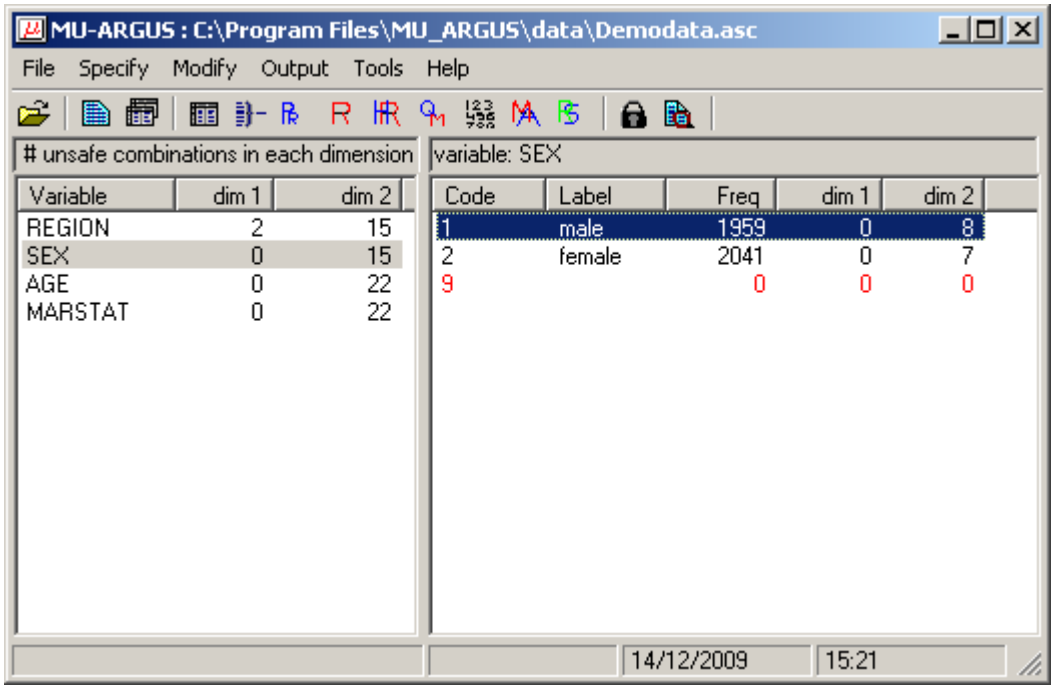


Figure 8. Output window for the individual risk methodology (2 sets of 2 key variables)

3.4.3 Menu Modify|Risk Specification: the Risk Chart window

As already mentioned, the risk table stored in μ-Argus contains the individual risk for each combination of key variables. This table cannot be accessed, but several aspects of the risk structure of the data can be inspected on request, either by selecting the menu Modify|Risk Specification, or by clicking the R icon on the toolbar. The **Risk Chart window** contains the available information on the risk structure of the file. If global recoding is applied to any of the key variables, the risk table is updated, and for assessing such protection procedure users must rely on a new inspection of the Risk Chart window.

3.4.3.1 Risk histogram

First of all, the **Risk Chart** window (Figure 9) illustrates the histogram of the individual risks, presented on a logarithmic scale. Note that if more than a single risk table is specified, users are requested to choose the graph to be inspected first.

The key variables used to compute the risk are specified on top of the graph. Finally, the option **Cumulative chart** transforms the histogram into a cumulative frequency graph.

3.4.3.2 Risk levels in the data before suppression

The risk chart window also contains information that refers to the file before suppression is applied to the data. The left-hand panel (Maximum levels in the file) shows the maximum individual risk in the file (5) and its re-identification rate (6),

expressed as 100*ξ %. The first of these quantities gives the highest individual risk that is attained in the original file, whereas the second represents the global risk of the file, i.e. the percentage (over the file size) of expected re-identifications in the file, would the file be released without suppressions. If global recoding is applied, this information is updated and the above mentioned values indicate the extent to which recoding affects the data. These values can therefore be used to assess whether the recoded file can be released without suppressions.

3.4.3.3 Setting the risk threshold to be used for local suppression

Before describing the rest of the window, we remark that, under the individual risk methodology, local suppression is driven by the value of the individual risk. This means that a rule, which is based on the risk, is applied to identify the *unsafe* records, i.e. the ones that need to be protected by suppression. Here the **unsafe records** are defined as those records whose risk is above a given **threshold**. Specification of this latter parameter is left to the user, who can proceed either directly, by selecting a **threshold** value for the **individual risk** r, or indirectly by imposing either a threshold (i.e. a maximum tolerable level) on the **re-identification rate** 100*ξ, or a target number of **unsafe records** n_u. In any circumstances, the left-hand panel, that contains information relative to the risks of the unprotected file, should be used as a benchmark in threshold setting.

First of all, the user can select the individual risk threshold manually, by using the **slider** (**1**) below the graph (see Figure 9). The **risk threshold** box **2**, the corresponding **number of unsafe records** **3** (i.e. the records whose risk is above the selected threshold) and finally the **threshold** on the **re-identification rate** (**4**) are automatically updated. A risk threshold can also be selected by using the tools in the **threshold setting** area of the risk chart window (central panel). If the user has already in mind a threshold on the individual risk, this value can be typed in directly in the **Risk threshold** box **2**. Clearly this value should be smaller than the maximum risk in the file, which is shown in **5**. As discussed in Section 3.3.3.7, the individual risk threshold can also be set through the re-identification rate; to this aim, the user must type a value in the **re-identification rate threshold** box **4**. Clearly this value cannot exceed the re-identification rate shown in the left hand panel (**6**). In all cases, the **Calc** button relative to the current box updates the quantities in the other boxes (number of unsafe records and re-identification rate threshold/risk threshold, respectively).

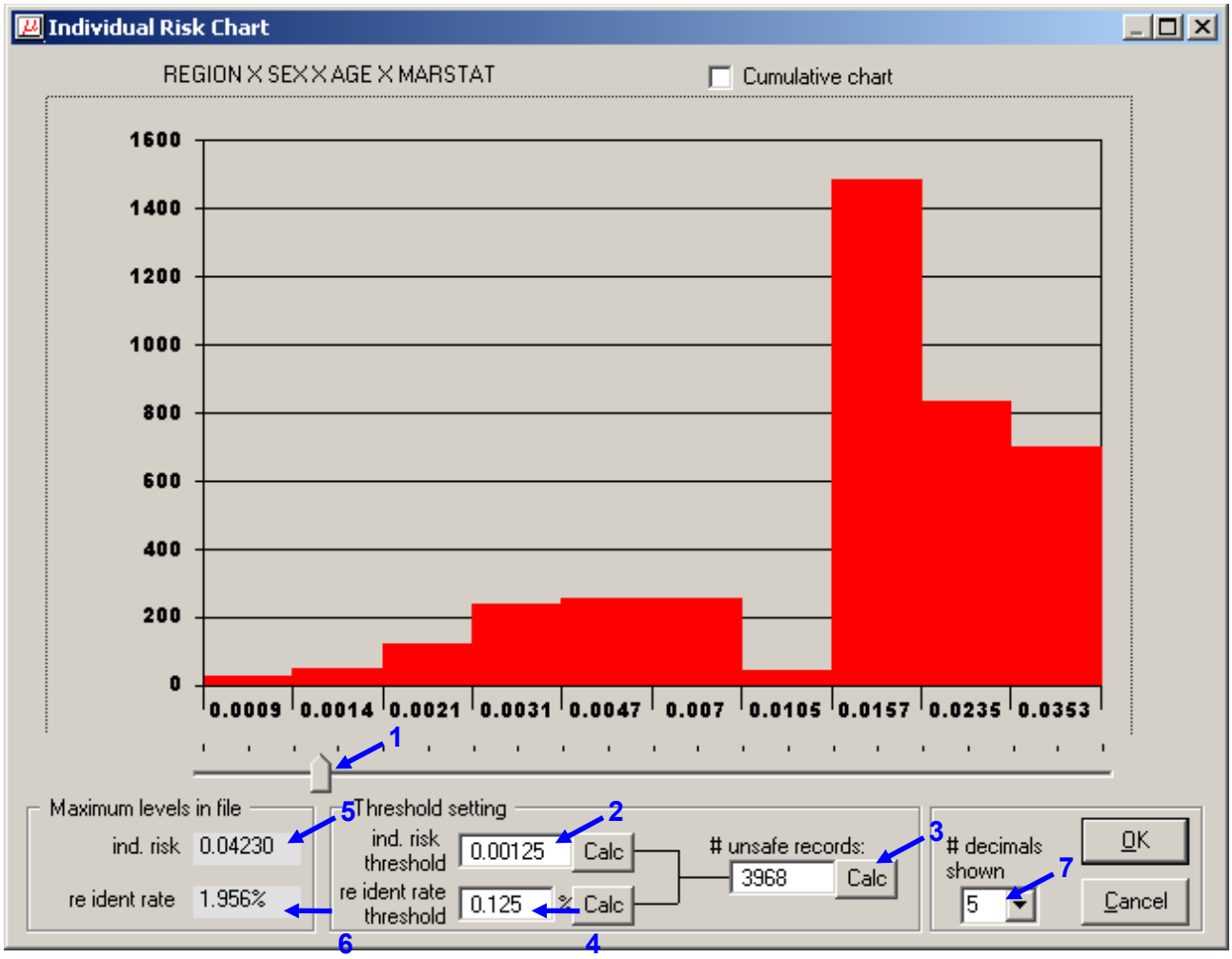


Figure 9. Risk chart for the set of key variables Region, Sex, Age and Marital status

For a given risk threshold r^* specified by the user, μ-Argus computes the corresponding number of unsafe records as $n_u = Card\{\ i\colon r_i \geq r^*\ \}$. However the estimated risks in fact form a discrete set of values; for this reason there is a whole set of threshold values compatible with the same number of unsafe records. Once typed a threshold r^* and determined the corresponding n_u, the user should compute the *actual* threshold level, e.g. the maximum *estimated* risk compatible with the given number n_u of unsafe records. This is obtained by pressing the **Calc** button **3**.

Of course the user can also determine the thresholds **2** and **4** on the individual risk and the re-identification rate, respectively, that correspond to the desired number of unsafe records; this must be typed in **3**. If necessary, the precision shown in the threshold boxes can be changed by selecting the appropriate number of digits in the right hand-side of the window (**7**).

Pressing the button OK sets the individual risk threshold at the *current* level. Recall that when the suppressed file is built, this level will be used to identify the unsafe records, i.e. the records to be protected. Choice of the threshold depends on the assumed scenario and on the uses of the file, e.g. whether it is intended as a microdata file for research or as a public use file. A subjective judgement of the availability of the external archive can also affect the threshold level. Eventually, the assessment of the appropriate threshold level for the data at hand, relative to the assumed scenario, is left to the person who is responsible for the release. In our

DEMODATA example, it could be considered safe to release a data file in which the expected number of re-identifications is certainly less than 5 expected re-identifications out of 4000 records. This corresponds to a threshold 100 ξ^* %= 0.125 %. μ-Argus calculates a corresponding risk threshold r^*= 0.00125, which means that in the released data the maximum probability of re-identification will be less than 1 out of 800 records[18]. However, prior to protection, the records whose risk is above this threshold are 3968; these are marked by the software as unsafe and undergo local suppression: applying suppression with this threshold would lead to a file in which nearly all records have at least one missing value, which cannot be considered a satisfactory compromise between protection and information loss.

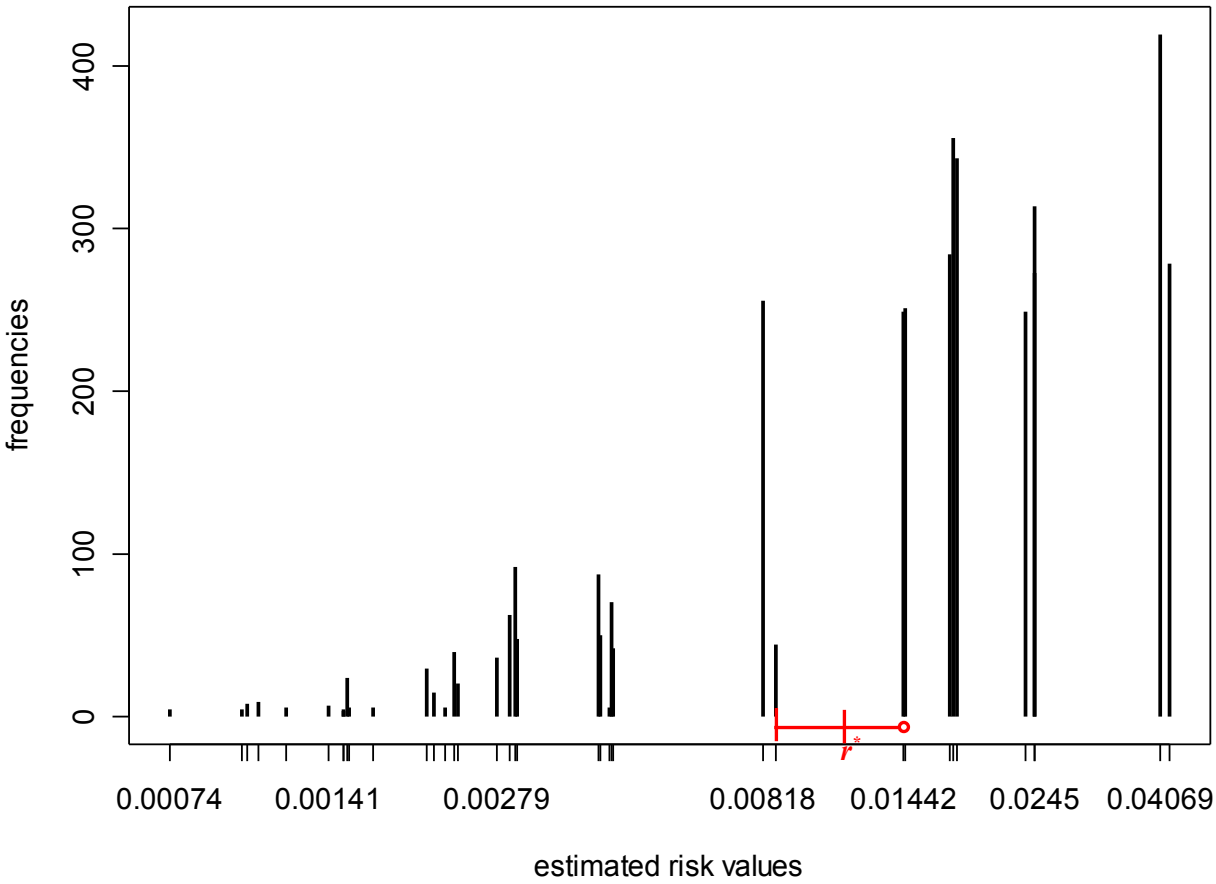


Figure 10. Graph of the risks (on a logarithmic scale) estimated for the DEMODATA file (key variables: Region, Sex, Age and Marital status; the interval of threshold values returning the same number of unsafe records as r^*.

As the above example shows, care should be taken of the number of unsafe records induced by the selected risk threshold. An exceedingly high number of unsafe

[18] In our example the *actual* risk values (i.e. the closest observed individual and global risk values that are below the nominal level) are 0.119 % for the re-identification rate threshold and 0.00119 for the individual risk threshold; higher protection is therefore achieved. To determine the actual levels, set the desired threshold, update the other values and use the calc button (**3**) of the number of unsafe records. In order to avoid that the number of unsafe records changes unexpectedly, merely as the effect of rounding, do not press the **Calc** button **2** again at this stage. Indeed in that case the software re-reads the value appearing in the risk box *using the digits shown*; this implies some rounding that might affect the number of unsafe records, that might change even if the risk threshold was not changed by the user.

records implies a high number of suppressions, that is, a highly corrupted microdata file. To ensure quality of the released data, it is advisable to apply *global recoding* and/or *bottom/top-coding* first. For the same example as in Figure 9 above, we recoded Age and Region to 6 and 18 classes, respectively. To this aim, the recode files REGION.GRC and AGE.GRC accompanying the μ-Argus DEMODATA file were used. Recall that after any manipulations of the key variables, the Risk Chart window is automatically updated; the Risk chart shown in Figure 11 can be accessed via the R icon, for instance.

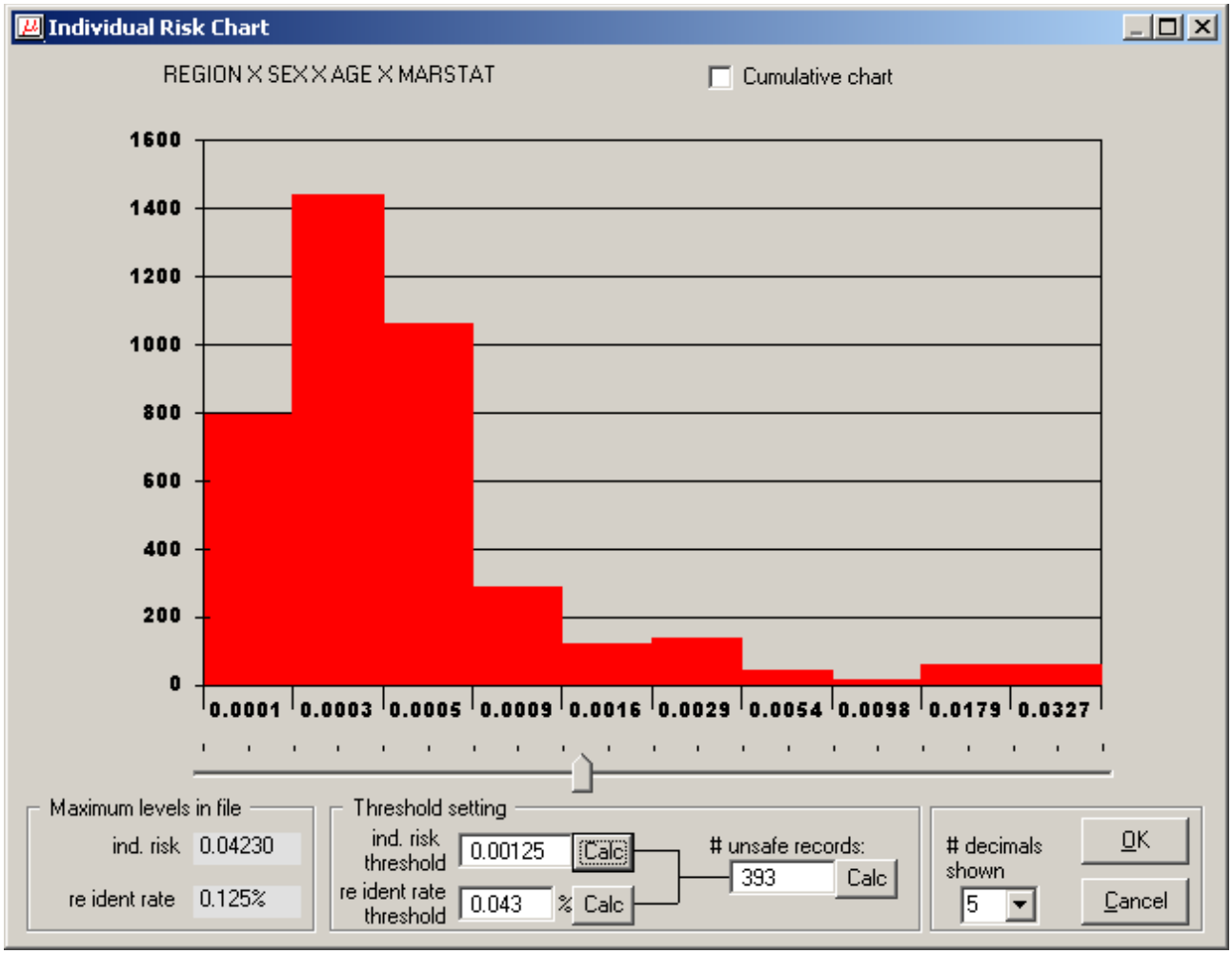


Figure 11. Risk chart for the set of key variables Region, Sex, Age and Marital status after recoding is applied to variables Region and Age.

Comparison of Figure 9 and Figure 11 illustrates the changes induced in the risk distribution by global recoding . Setting the same risk threshold as before, we notice that recoding has considerably lowered the number of unsafe records, that are now less than 400. Of course, the decrease in the number of unsafe records depends on how recoding is applied, i.e. on the number of collapsed categories and their frequencies. Compared to the previous example, records are now allocated to cells of a broader contingency table; consequently the sample cell frequencies are increased and the risk is lowered for most records, not only the unsafe ones. The global risk of the unprotected file has also sensibly decreased (0.125%, or 5 expected re-identifications in the file, instead of 1.956%, i.e. about 78 expected re-identifications in the raw data).

In practice a combination of approaches is usually adopted, as the example has made clear: so far, Istat has been applying first global recoding; the re-identification rate of the file before suppression, the maximum risk level and the whole individual risk graph help the user in assessing whether the solely recoded file can be considered safe; current practice at Istat consists in first recoding the variables to the finest detail at which the estimates retain their planned precision. Some of the key variables –usually geographical variables and age- are also considered for further recoding. Numerical key variables may also be treated by top/bottom coding. If the risk properties of the file do not meet the required standard, local suppression, based on the individual risk of re-identification, is applied. Clearly the number of suppression induced by the threshold should always be inspected to decide whether other solutions (further recodings, suppression of variables from the file…) should be considered. Such a strategy aims at maintaining the quality of the file, while guaranteeing confidentiality.

3.4.4 Menu Modify|Household Risk Specification: the Household Risk Chart window

When a household identifier is detected in the metadata, and the risk methodology is chosen, the household risk described in Section 3.3.4 is automatically computed. The Modify menu (or the button) permits to access the household risk window. Its aspect is very similar to that of the Individual Risk Chart window.

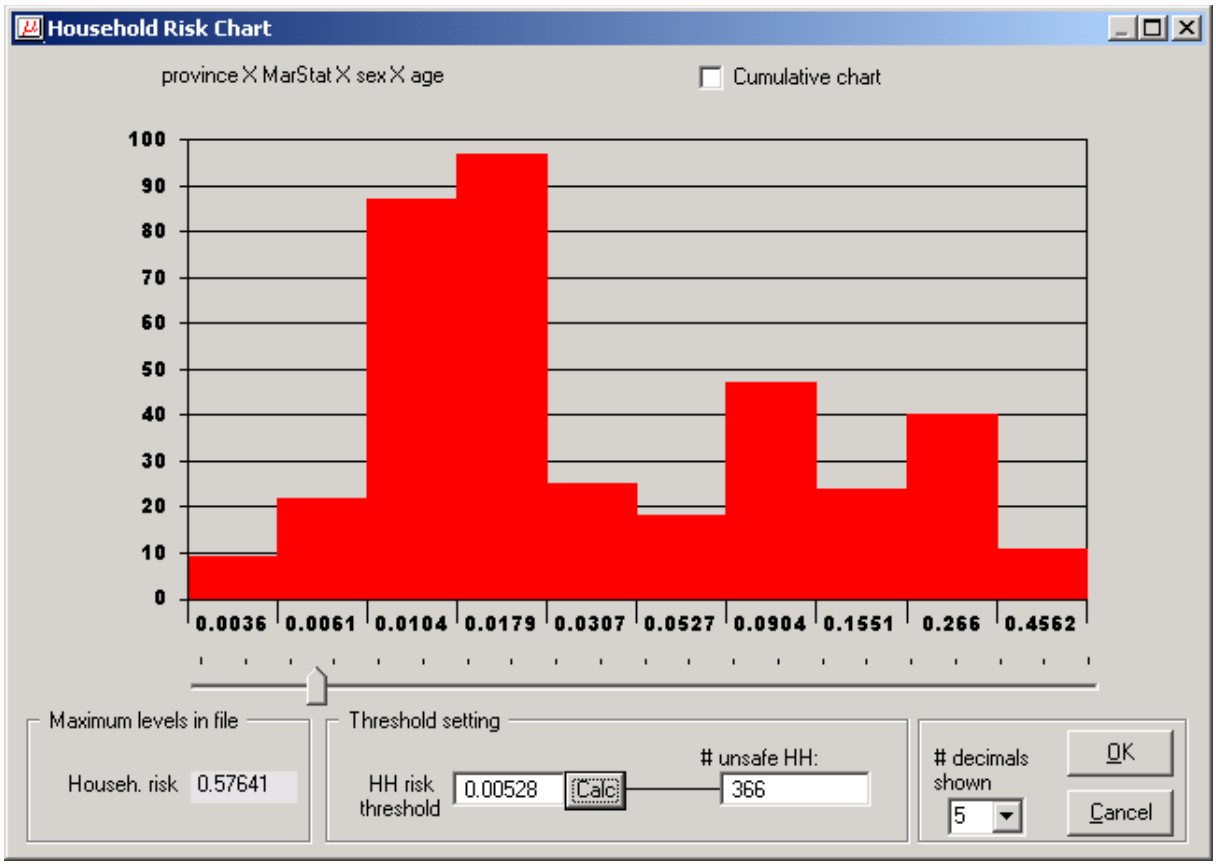


Figure 12. Household risk window for the HHDATA demonstration file

Using the demonstration file SAMHH.ASC, we follow the usual procedure to create a risk table having the following key variables: province, MarStat, sex, age. The corresponding household risk chart window is shown in Figure 12. By this window the user can perform an assessment of the (household) risk of the data and identify the unsafe records as discussed in the next section.

3.4.4.1 Setting the household risk threshold and identifying unsafe records

In a file having a household structure, the unsafe *records* are defined in two steps: first a threshold on the *household* risk is selected; in this way *households* are defined as safe or unsafe. However, depending on the values of the key variables, within the same household there will be high risk as well as low risk records. As the household risk is the composition of the individual risks of the members of the household, to decrease the household risk it will be most effective to protect the records whose individual risk is highest. Therefore the second step in the procedure is to define a rule that permits to identify, within the unsafe households, the records to be suppressed. These will be referred to as the unsafe (or at risk) records. By this rule, in fact only a *subset* of the records belonging to unsafe households will be classified as unsafe. As discussed in Section 3.3.4.2, our rule to detect the unsafe records relies on a condition that is sufficient to ensure that, after protection, a household is safe for a given threshold r^{h*} on the household risk. Such condition is that all records in the household do not contribute more than $(1/d)$-th of the household risk, d being the household size. This in fact amounts to using different thresholds for different household sizes.

To illustrate the procedure to identify unsafe records, we refer to the SAMHH data example. *For illustration purposes only*, we position the slider at a threshold r^{h*}=0.13431. This value identifies 71 unsafe households, three of whom are shown in Figure 13. Here for instance the first household has three members, of whom only the second has high individual risk. For a household risk threshold r^{h*}=0.134310 as in the example, records of the first household are defined *safe* whenever their individual risk is below 0.04477; only the second record in the first household is therefore set to unsafe. Likewise, in the third household, the first two records are at risk for the given threshold r^{h*}=0.134310. For households of size 1, like the third household in Figure 13, this procedure induces no difference between the household and the individual risk threshold.

```
742221 32421 162 42      0.040238671962 0.199476456345
742221 11192   0 42      0.146135380971
742221 12162   0 42      0.023163578416
751111 31882   0 88      0.238164748136 0.238164748136
761111 21331 572 33      0.075354624319 0.198838280708
761111 22331 110 33      0.075354624319
761111 12 72   0 33      0.031978745825
761111 11 52   0 33      0.031978745825
```

Figure 13. Data selected from the demonstration file samhh.dat. Each records is sided by the individual and household risk. The household risk is reported only for the first record in the household.

Once a household risk threshold r^{h*} is selected in the Household Risk Chart window, the number of unsafe households and records, respectively, is shown in the proper box. As explained in the previous example, the number of unsafe records is generally smaller than the total number of records belonging to unsafe households.

3.4.4.2 Validity of the assumptions that define the scenario for estimating the household risk

The disclosure scenario defined in Section 3.3.4.1 for the release of files of households assumes that the intruder's archive refers to individuals and does not contain information about the household structure. Depending on the information provided by the public archives, sometimes it might be more realistic to assume that the re-identification attempt also makes use of household variables such as the size of the household and/or the household type. In this circumstance we recommend to use these as additional key variables in computing the risk, so as to allow for the additional information concerning the household. If not directly available in the data, both variables can be computed from the file.

If instead it is deemed more appropriate to define a scenario under which the intruder has access to an archive of households, each containing as many individual records as there are members in the household, then the definition of household risk is not fully appropriate. In this circumstance, including the household size and the household type in the key variables permits to approximate the appropriate household risk, i.e. allows for part of the dependence between records in the same household.

3.4.5 Output: Make protected file window

Once the individual and/or global risk thresholds, and the corresponding number of unsafe records are deemed appropriate, the output file (safe file) can be produced. From the menu Output select Make protected file. The window Make protected file pops up (see Figure 14). This window looks the same for all the protection methods offered by μ-Argus. In the application of the individual risk methodology, the left panel shows the key variables only; as partial suppression only affects these variables. If this is reasonable, users can ask for no suppressions, otherwise a criterion to place missing codes in the data must be chosen. μ-Argus has two options, either entropy or suppression weights. With the latter option, users can adjust the suppression weights manually as it happens with the traditional (Dutch) technique. Introducing different weights can be useful to give different suppression penalties for different key variables: for instance, users can assign sex the maximum weight (100) so as to discourage suppressing that variable, as long as it is possible. With both options, the suppression criterion used by μ-Argus is based on an optimisation algorithm. Missing codes are placed so as to minimise a measure of information loss (either entropy or suppression weight). In any case, the effect of suppression is to lower the individual risk of the protected records; after protection , this will be below the selected threshold r^*.

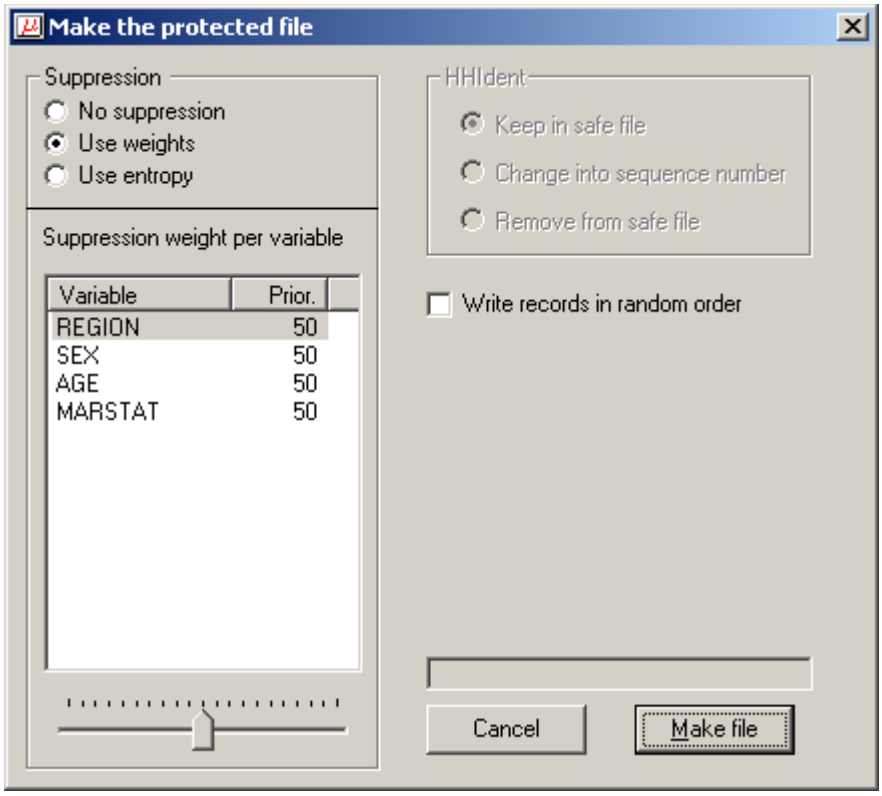


Figure 14. Make protected file window

Pressing Make file will store the safe file. A report is produced, that details all the procedures that lead to the safe file. Besides information on the original and protected file, the report details the main aspects of the protection: a list of the key variables used in the model and the risk table, the list of recodings, when global recoding has been applied, detailed for each recoded variable, the maximum risk levels in the output file, i.e. the individual risk threshold, the global risk threshold, and, if applicable, the household risk threshold; finally, the marginal number of suppressions per variable needed to produce a data file that is safe for the given threshold levels.

The report also contains a record description of the output We next show an excerpt from one such report for the example at hand (for the DEMODATA file, that does not carry a household structure).

The final results can be assessed by inspection of the **suppression overview** in the report. If the number of suppressions per variable is not satisfactory, the user may want to *recode* selected variables; in this case, go to **Modify|Global Recode** and inspect the new Risk Chart, possibly selecting a new risk threshold. The process may be iterated until satisfactory results are produced.

The risk threshold, *if appropriate,* can also be modified; in this case, go back to the Risk Chart.

References on the risk estimation

Abramowitz, M. and Stegun, I. A. (1965). Handbook of Mathematical Functions, Dover, New York.

Benedetti, R. and Franconi, L. (1998). Statistical and technological solutions for controlled data dissemination, *Pre-proceedings of New Techniques and Technologies for Statistics*, 1, 225-232.

Benedetti, R. and Franconi, L. and Capobianchi, A. (2003). Individual Risk of Disclosure Using Sampling Design Information. *Contributi Istat* n.14-03.

Bethlehem, J., Keller, W. and Pannekoek, J. (1990). Disclosure control of microdata, *Journal of the American Statistical Association* 85, 38-45.

Carlson, M. (2002). Assessing microdata disclosure risk using the Poisson-inverse Gaussian distribution. *Statistics in Transition* 5, 901–925.

Chen, G. and Keller-McNulty, S. (1998) Estimation of identification disclosure risk in microdata. *Journal of Official Statistics* 14, 79–95.

Deville, J.C. and Särndal, C.E. (1992). Calibration estimators in survey sampling, *Journal of the American Statistical Association* 87, 367–382.

Di Consiglio, L., Franconi, L. and Seri, G. (2003). Assessing individual risk of disclosure: an experiment, *Proceedings of the Joint ECE/Eurostat Work Session on Statistical Data Confidentiality*, Luxembourg.

Duncan, G.T. and Lambert, D. (1986). Disclosure-limited data dissemination (with comments), *Journal of the American Statistical Association* 81: 10-27.

Elamir, E. and Skinner, C. (2004). Modeling the re-identification risk per record in microdata. *Technical report*, Southampton Statistical Sciences Research Institute, University of Southampton, U.K.

Fienberg, S.E. and Makov, U.E. (1998). Confidentiality, uniqueness and disclosure limitation for categorical data. *Journal of Official Statistics*, 14, 385-397.

Franconi, L. and Polettini, S. (2004). Individual risk estimation in μ-Argus: a review. In: Domingo-Ferrer, J. (Ed.), *Privacy in Statistical Databases*. Lecture Notes in Computer Science. Springer, 262-272

Lambert, D. (1993). Measures of disclosure risk and harm. Journal of Official Statistics 9, 313–331.

Polettini, S. (2003a). A note on the individual risk of disclosure. *Contributi Istat* n.14/2003

Polettini, S. (2003b). Some remarks on the individual risk methodology. In: *Proceedings of the Joint ECE/Eurostat Work Session on Statistical Data Confidentiality*. Luxembourg.

Polettini, S. (2004). Revision of Deliverable 1.2-D3 "*Guidelines for the protection of social micro-data using the individual risk methodology*" by S. Polettini and G. Seri, available at http://neon.vb.cbs.nl/CASC

Polettini, S. and Stander, J. (2004). A bayesian hierarchical model approach to risk estimation in statistical disclosure limitation. In: Domingo-Ferrer, J. (Ed.), *Privacy in Statistical Databases*. Lecture Notes in Computer Science. Springer, 247-261

Rinott, Y. (2003). On models for statistical disclosure risk estimation. In: *Proceedings of the Joint ECE/Eurostat Work Session on Statistical Data Confidentiality*. Luxembourg.

Skinner, C.J. and Holmes, D.J., (1998). Estimating the re-identification risk per record in microdata. *Journal of Official Statistics*, 14, 4, 361-372.

Willenborg, L. and de Waal, T. (1996). *Statistical Disclosure Control in Practice*. Lecture Notes in Statistics, 111, New-York: Springer Verlag.

Willenborg, L. and de Waal, T. (2001). *Elements of Statistical Disclosure Control*, New York: Springer-Verlag.

3.5 Information loss

A measure for disclosure risk such as a thresholding rule or the more sophisticated individual risk approach can be used as a criterion to distinguish between safe and unsafe microdata. If unsafe microdata are going to be transformed into safe microdata it is necessary to have a measure of information loss at one's disposal. This is used to limit the amount of "damage" done to the data when they are being modified. In practice, if the modifications are being carried out interactively by a data protector, this person is likely to use a crude and intuitive measure of information loss. However, if the modification is (partly) done automatically using μ–ARGUS it is necessary that the package uses some measure for the information loss. In case of applying local suppressions only, μ–ARGUS simply counts the number of local suppressions. The more suppressions the higher the information loss. In case of automatic global recoding μ–ARGUS uses an information loss measure that uses the following parameters: a valuation of the importance of an identifying variable (according to the data protector), as well as a valuation of each of the possible predefined codings for each identifying variable.

Both global recoding and local suppression lead to a loss of information, because either less detailed information is provided or some information is not given at all. A balance between global recoding and local suppression has to be found in order to make the information loss due to SDC measures as low as possible. It is recommended to start by recoding some variables globally until the number of unsafe combinations that have to be protected by local suppression is sufficiently low. The remaining unsafe combinations can then be protected by suppressing some values.

3.6 Sampling weights

So far we only discussed the most direct form of re-identification, namely that of an intruder recognising an individual. Sometimes a more indirect form of disclosure is possible. This is for instance the case if a microdata file contains certain explicit information that allows an intruder to deduce other information that a data releaser would not like to release. It is not so much the danger of a privacy threat that drives the prevention of this type of disclosure, but rather the prevention of an embarrassing situation for the data releaser. For it is somewhat embarrassing if information can be derived from the data that is not supposed to be available. An example of this is postsampling weights. Such weights are often included in a microdata file so as to correct for all sorts of defects in a sample due to selective response and over- or undersampling of certain subpopulations. However, knowledge of the procedure to calculate such weights, together with some population data (stratum frequencies) can be supposed to be generally known. With this information an intruder might in some cases be able to re-identify the stratum to which a respondent belongs. Knowing the stratum means in fact that certain attributes (that define the stratum) are known to an intruder. This could for example be a region. If it were the policy of the data releaser not to release any regional data in the file, this would be an embarrassment. This can be avoided in various ways. It is possible to draw a suitable subsample (records) from the microdata set, such that the weights for this new sample are not very different. It is also possible to add some noise to the weights, in order to hamper the stratum re-identification procedure. In μ–ARGUS it is possible to add noise to the weights.

3.7 Household variables

Some identifiers necessarily yield the same scores for all members of the same household. Such variables are called household variables. Examples of such variables are "the size of the household" and "the occupation of the head of the household". If a household variable determines the households in the population uniquely it is called a household key. Household variables in a microdata set containing records pertaining to different persons in a household, some of whom are represented in this file, may allow an intruder to group the records of persons belonging to the same household. This is an extra disclosure risk that threatens a microdata set containing such variables. The increased risk follows from the fact that households might be more easily recognisable than individual persons. μ–ARGUS is able to deal with household variables. This means that if a value of a household variable in a record is suppressed, then the corresponding values are suppressed in the records referring to the other members of the same household.

3.8 Blocking procedures

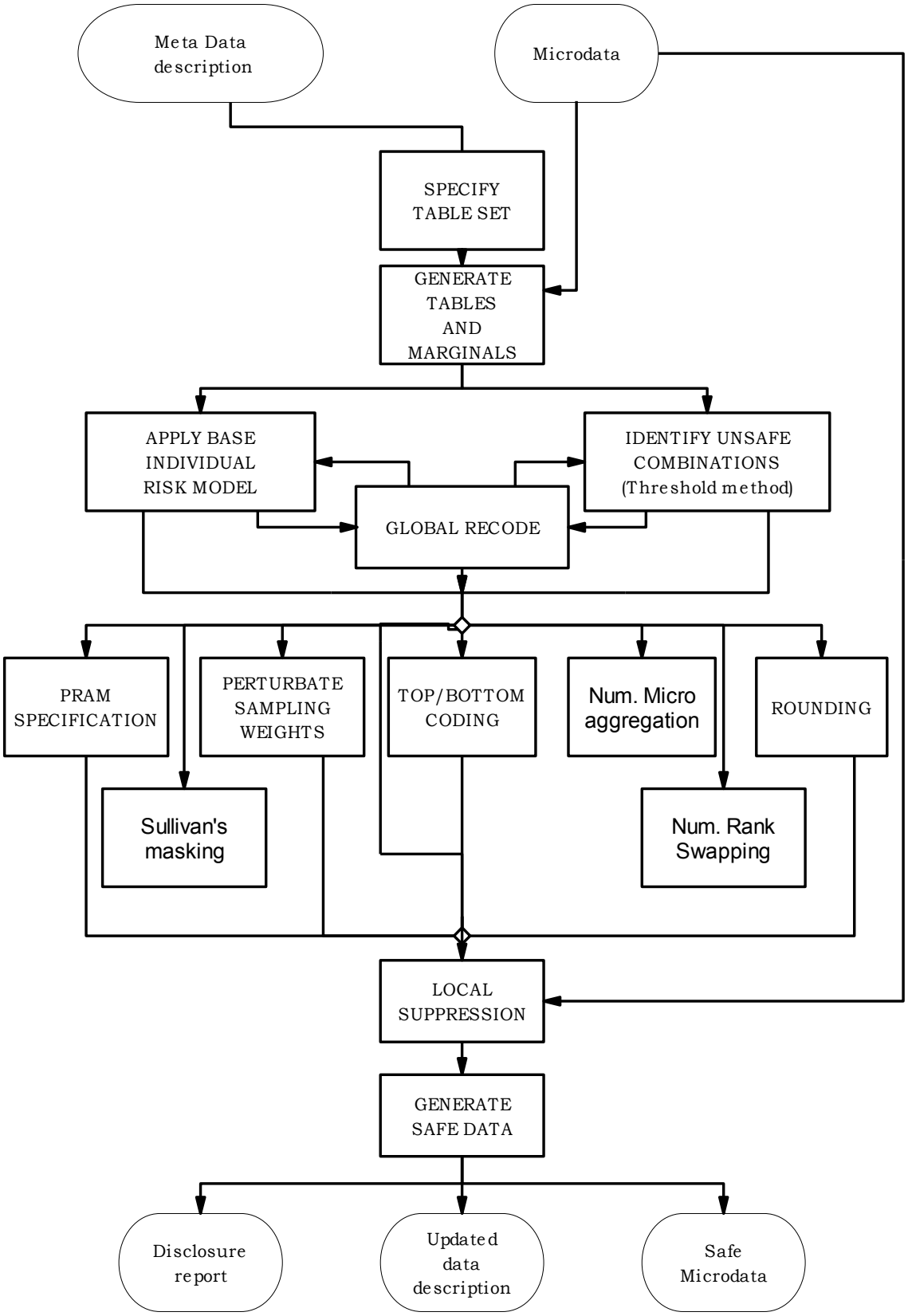
Several procedures in μ-ARGUS may become very time-consuming when applied to a complete large dataset. As a deliverable of the ESSNet project a report is available on blocking procedures, see Domingo-Ferrer (2008).

In μ-ARGUS a first application is introduced. For certain time-consuming procedures like microaggregation, rank swapping and record linkage a blocking variable can be selected. The datafile will then be split in several sub-sets. The procedures will then be applied on the sub-sets and after that the outcome is combined for the whole dataset. Of course the selection of the blocking variable is important. Some attention should be played to the selection. On eth one hand side the data should be split into manageable subsets and the division should be meaningful with respect to the procedure applied.

References on blocking methods

Domingo-Ferrer (2008) *Blocking Methods for Microdata Protection,* ESSNet deliverable, available at http://neon.vb.cbs.nl/casc/essnet/essnet_sdc_blocking_deliverable.pdf

3.9 Functional design of μ-ARGUS



4 A TOUR OF μ-ARGUS

This section will give the reader a brief introduction through the use of μ–ARGUS. Some Windows experience is assumed. In this section we will use a demo-dataset called DEMODATA, which has been supplied with the μ–ARGUS software. In section 4 a more systematic description of the different parts of μ–ARGUS will be given. Please note that all options in μ-ARGUS can be accessed via the drop-down menu, but also via the toolbar. In this tour we will use the drop-down menu.

4.1 Preparation

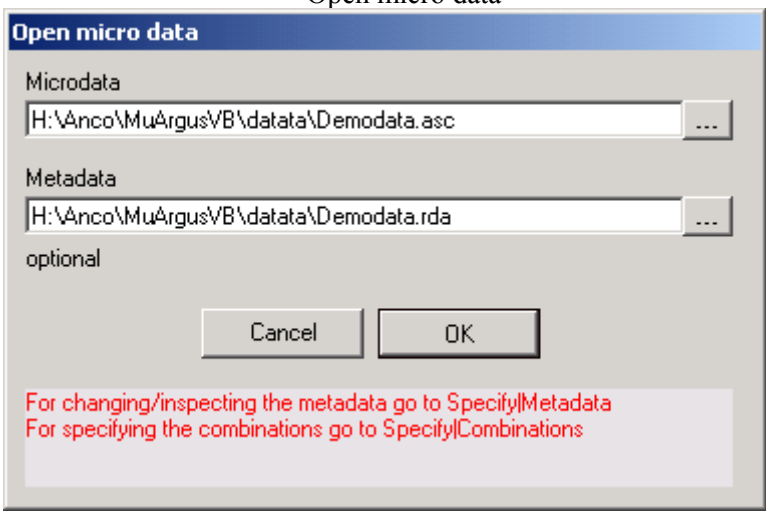
When you double click the μ–ARGUS icon the program starts. To start the disclosure control with μ–ARGUS you need to have a microdata file and the metadata describing this microdata file. Three options are now available for microdata file format:

- fixed format (as previously),
- free format (with a defined separator),
- free format with variable names in the first line (called FreeWithMeta format in μ–ARGUS). A format very convenient for e.g. SAS-users.
- SPSS system files

Here we restrict our attention to fixed format input. The other formats are discussed in Section Error: Reference source not found.

In the methods presented in Section 3, the microdata file must be a fixed format ASCII file. If you click (File|Open microdata) you can specify the name of the microdata file. The program assumes the extension .asc, but you can use your own extension.

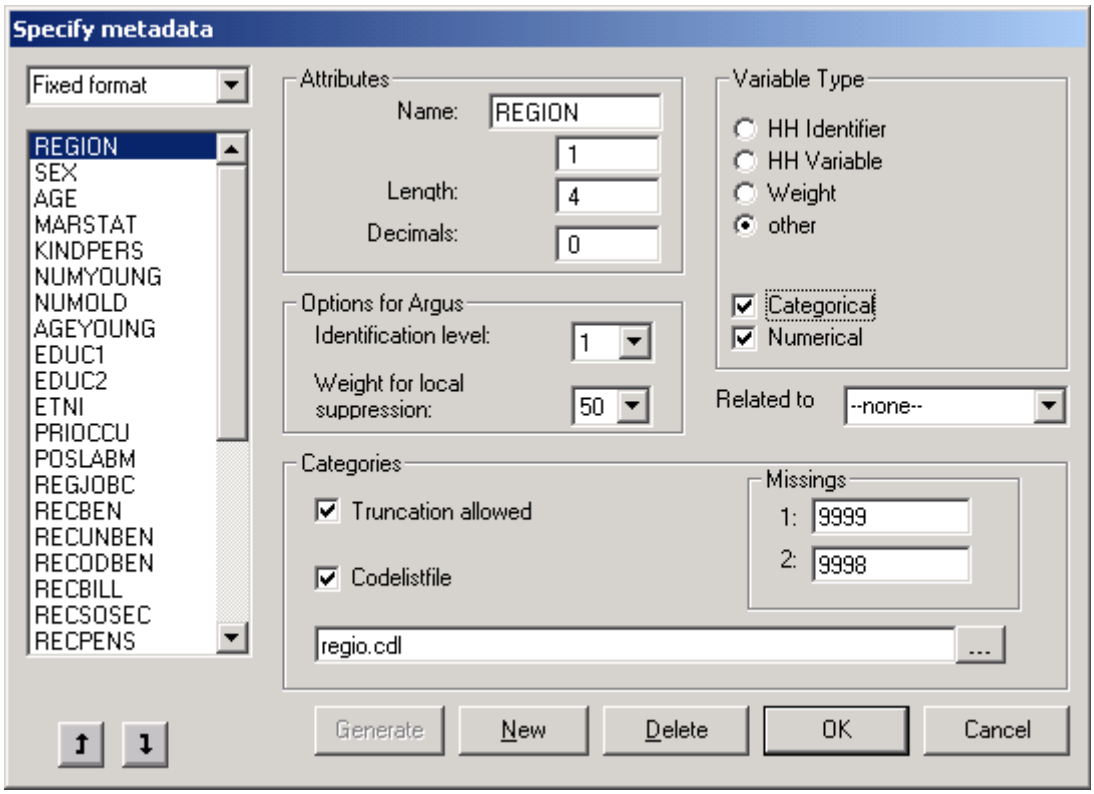
Open micro data



The metadata describing this ASCII file is stored in a separate file. For this file the program assumes the extension .rda (Record Description for Argus), but you can select another extension. If no metadata file is specified, the program has the

facility to specify the metadata interactively via the menu option (Specify|Metadata). In this section, we will give a description of how the use of (Specify|Metadata) results in a display of the information in the metadata file for μ–ARGUS. When you enter or change the metadata interactively using μ–ARGUS the option (Specify|Metadata) will bring you to this screen:

Specify metadata



In a first tour you should leave the fields as they are, but we will explain here the meaning of these fields.

In the left pane of the 'Specify metadata' window the available variables of the dataset are shown. In the other fields the various attributes, relevant for the disclosure control process are shown.

- The starting position in each record of the datafile will be familiar.
- The Variable Type:
    - HH Identifier: The unique identifier of a household
    - HH Variable: A variable that by nature has the same value for each member of a household
    - Weight: The variable is a sampling weight
    - Categorical: A categorical variable can be used as a spanning variable in a table. For μ-ARGUS this variable can be defined as identifying
    - Numerical: A numerical variable can be used for top/bottom coding, microaggregation and rounding
- The identification level is an option to easily generate the set of tables to be inspected in the disclosure control process. The contents of these sets are based

on certain rules used by Statistics Netherlands, but they could easily be used in other situations as well. This procedure is further explained in the next section

  - **0**: an individual cannot be identified by this variable and it will not play a role in the disclosure control process.
  - **1**: the variable is most identifying.
  - **2**: the variable is more identifying.
  - **3**: the variable is identifying.

- The weight for local suppression. When at the end of a μ-ARGUS run the remaining unsafe combinations are suppressed by local suppression, one of the options is to use the user-supplied weights to select the variables that will be suppressed. These weights are specified here.
- The codelists of the variables used to span the tables is always generated by μ-ARGUS itself. However the user can supply the name of a codelist file (see example below in the metadata file example). The labels in this codelist file are then used when displaying information on this variable. Besides the name of the codelist file it is possible to indicate whether truncation is a feasible way of recoding. This is the case when the codelist is hierarchical.
- One or two missing values can be specified per variable. Missing values play a specific role in the SDC-process, as missing values will be imputed when local suppression is applied. Note that the weight variable cannot have a missing value, but categorical variables need to have at least one missing value.

The meta data is stored in an RDA-file. A further description of this file can be found in section 6.1

4.1.1 Specify combinations.

When the metadata is ready, the set of combinations to be inspected by μ–ARGUS can be specified. Either you specify the tables manually or you use one of the two rules to generate this set. If you select the tables manually you select the variables in the left pane and move them to the middle pane (with the '>'-button). If the table is ready and the appropriate threshold has been specified you add the table to the set of tables in the right pane with the '->'-button. Each table can have a different threshold.

If you use the 'automatic specification of tables' , you will have two options, based on the rules used in Statistics Netherlands, 1: using the identification level and 2: all tables up to a given dimension.

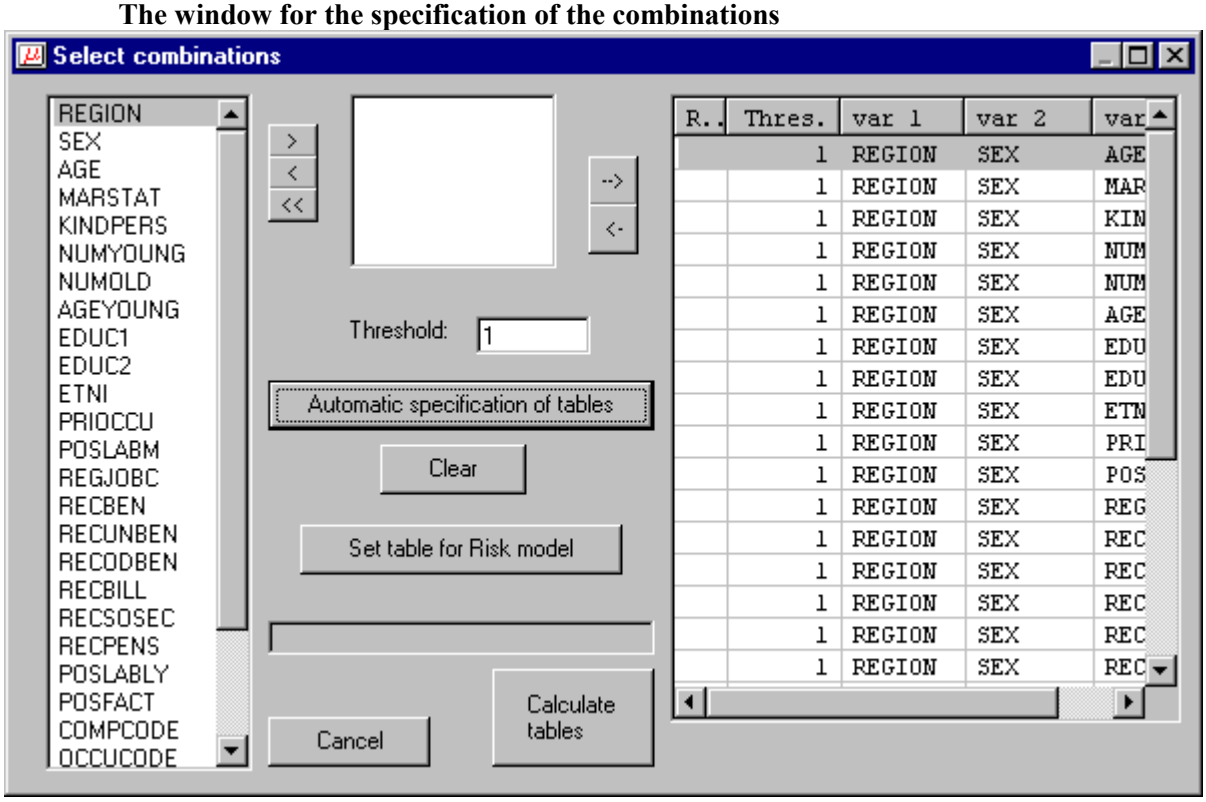
1. In the first case the set of tables is constructed as follows: For each variable the parameter identification level must be specified. The selection of variables is restricted to those variables where the identification level is >0. Each variable with a level = 1 is combined with each variable of level ≤ 2 with each variable of level ≤3 etc. At Statistics Netherlands this rule is used to produce microdata files for researchers (MUC). For all tables the same threshold must be specified.
2. In the second case all tables up to a given dimension are used. The selection of variables is restricted to those variables where the identification level is >0. For each dimension a different threshold can be specified. At Statistics Netherlands this rule is used to produce public use files (PUF).

When the set of tables is constructed using the generator, it is still possible to make adjustments. You can add additional ones or delete certain tables. If a large number of tables is being generated due to the specific parameters μ-ARGUS will ask whether it should proceed or not.

Independent of the way in which the set of tables is constructed, the threshold for the tables must be specified - separate thresholds for each number of dimensions in the table (1d, 2d, etc). This threshold is the maximum value of a cell in a table, which is still considered unsafe. Above this threshold a cell is considered safe. In case of a sample survey this 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 by the user.

If the new risk model is used, you select a table for this model by pressing the button 'set table for risk model'. A description of this model can be found in section 3.2.3. A restriction is that there cannot be an overlap between the tables used for the classical threshold method and the new risk-model. Mixing the basic model and the new risk approach makes no sense. Therefore the overlapping tables will be removed automatically.

The window for the specification of the combinations



When you are satisfied with the specified set of tables you press the button 'Calculate tables'.

This will be done in three phases. In the **first phase** μ–ARGUS will inspect the microdata file and construct lists of existing codes for the active variables. These lists are constructed independently from the specified codelists. This is done in order to prevent from problems arising when some existing codes might be missing in the codelists. When the codelists have been computed, μ–ARGUS will in the **second phase** calculate the tables to be inspected. In the final **third phase** the

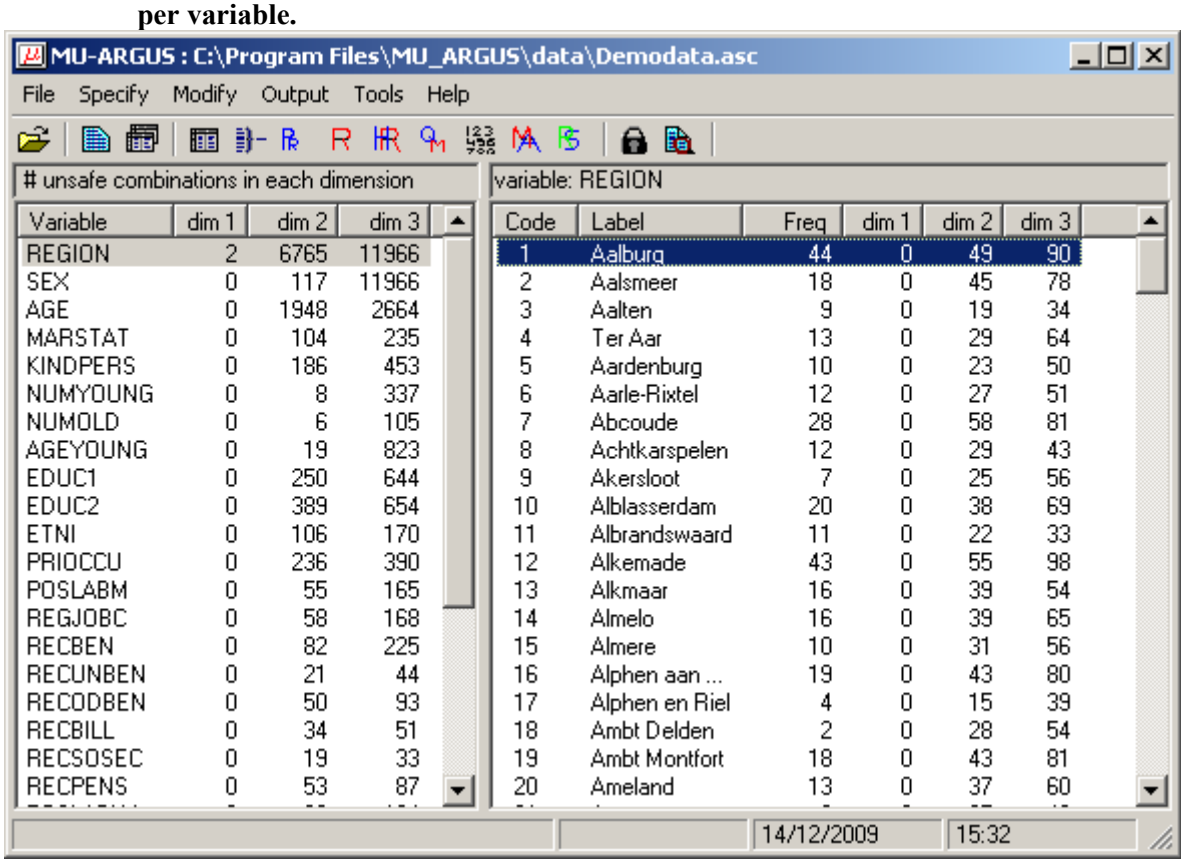
marginals of the tables will be calculated. The progress is shown by the progress bar.

After this you are ready to start the disclosure control process.

4.2 The process of Disclosure Control

When all the preparations have been made the actual process of Disclosure Control can start. The two main actions that can be performed are Global Recoding and Local Suppression. Additionally you can modify numerical variables by 'top/bottom coding', 'rounding', 'numerical Micro Aggregation', 'numerical Rank Swapping'. The user can select and perform manually the Global Recodings (via Modify|Global Recode). The central μ-ARGUS window is the main window where for each variable the number of unsafe combinations is shown by dimension of the table. The left hand window displays the number of unsafe cells in one-dimensional tables involving the named variable; then the same information is given for tables of greater dimension up to the highest dimension created by the user. For the selected variable in the left window, the number of unsafe combinations for each category of the selected variable is shown in the right hand window. The higher the number of unsafe combinations for a variable the more likely it is that the variable needs to be globally recoded in order to achieve a safe microdata file.

The main menu for μ-ARGUS, showing the number of unsafe combinations per variable.



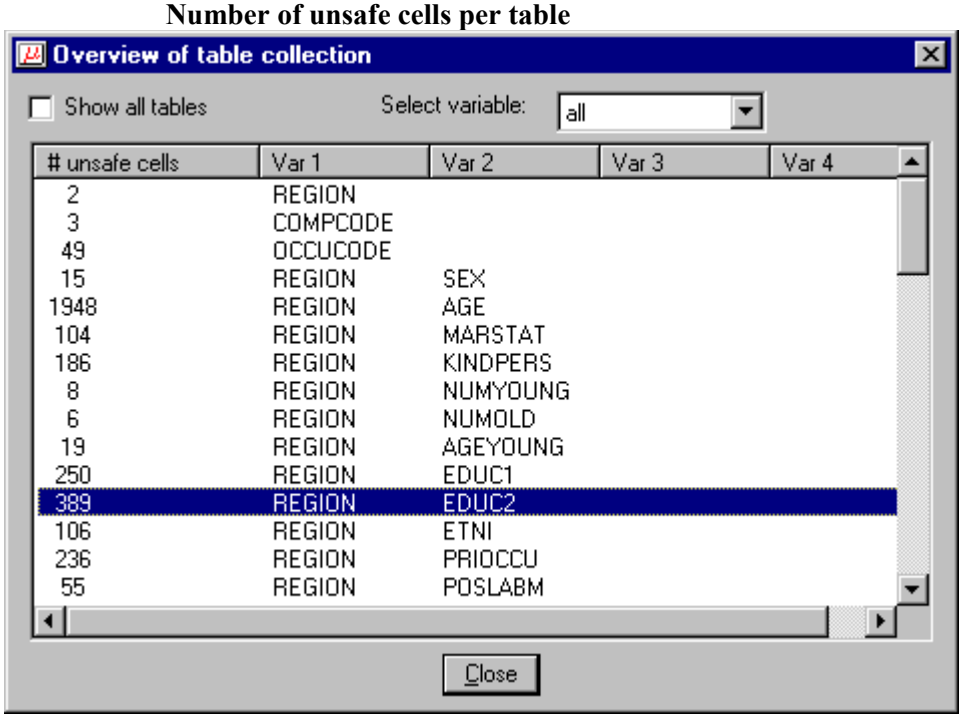
Variable	dim 1	dim 2	dim 3
REGION	2	6765	11966
SEX	0	117	11966
AGE	0	1948	2664
MARSTAT	0	104	235
KINDPERS	0	186	453
NUMYOUNG	0	8	337
NUMOLD	0	6	105
AGEYOUNG	0	19	823
EDUC1	0	250	644
EDUC2	0	389	654
ETNI	0	106	170
PRIOCCU	0	236	390
POSLABM	0	55	165
REGJOBC	0	58	168
RECBEN	0	82	225
RECUNBEN	0	21	44
RECODBEN	0	50	93
RECBILL	0	34	51
RECSOSEC	0	19	33
RECPENS	0	53	87

Code	Label	Freq	dim 1	dim 2	dim 3
1	Aalburg	44	0	49	90
2	Aalsmeer	18	0	45	78
3	Aalten	9	0	19	34
4	Ter Aar	13	0	29	64
5	Aardenburg	10	0	23	50
6	Aarle-Rixtel	12	0	27	51
7	Abcoude	28	0	58	81
8	Achtkarspelen	12	0	29	43
9	Akersloot	7	0	25	56
10	Alblasserdam	20	0	38	69
11	Albrandswaard	11	0	22	33
12	Alkemade	43	0	55	98
13	Alkmaar	16	0	39	54
14	Almelo	16	0	39	65
15	Almere	10	0	31	56
16	Alphen aan ...	19	0	43	80
17	Alphen en Riel	4	0	15	39
18	Ambt Delden	2	0	28	54
19	Ambt Montfort	18	0	43	81
20	Ameland	13	0	37	60

The unsafe combinations are calculated by checking the threshold value in each combination. Please note that if e.g. a three-dimensional combination is checked also the one and two dimensional marginal combinations are checked. In this window also the scores of these marginal tables are shown.

As the unsafe combinations are calculated for the tables specified, it is possible to inspect the number of unsafe cells per table via the menu option (Modify|Show tables). The following window is shown:

Number of unsafe cells per table

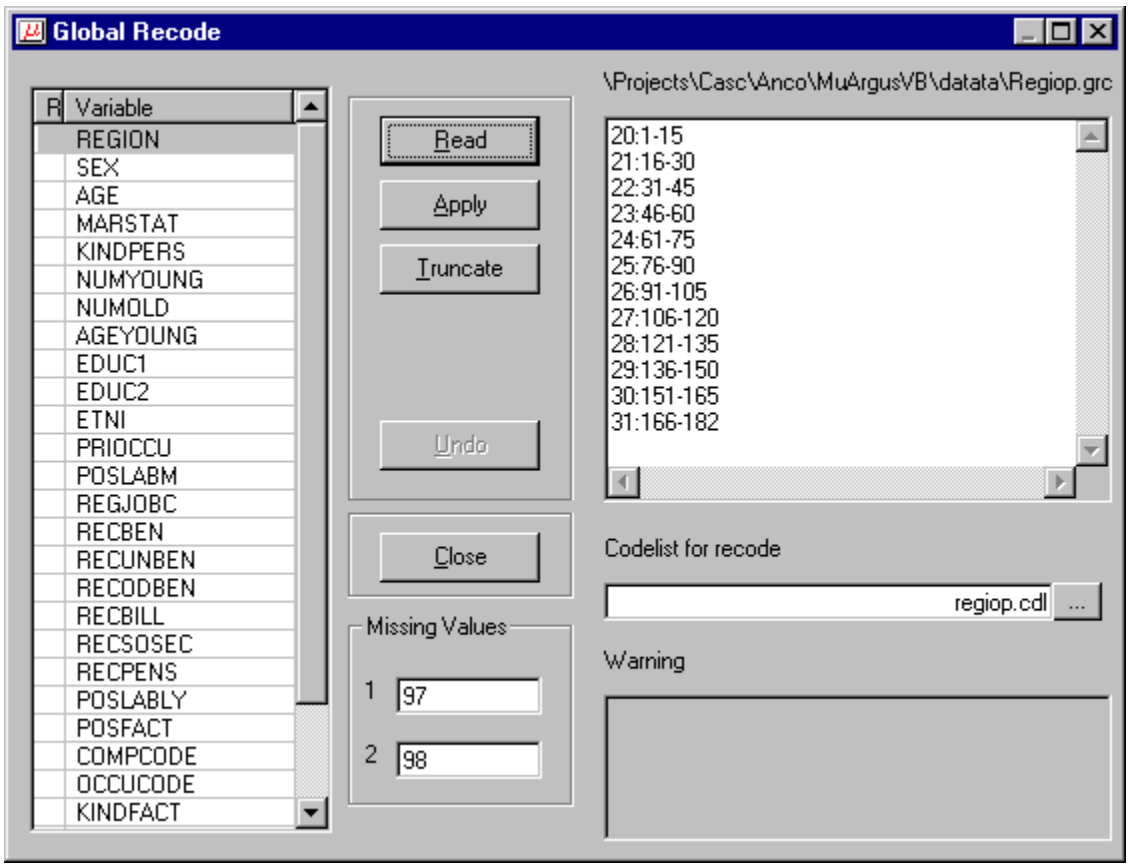


# unsafe cells	Var 1	Var 2	Var 3	Var 4
2	REGION			
3	COMPCODE			
49	OCCUCODE			
15	REGION	SEX		
1948	REGION	AGE		
104	REGION	MARSTAT		
186	REGION	KINDPERS		
8	REGION	NUMYOUNG		
6	REGION	NUMOLD		
19	REGION	AGEYOUNG		
250	REGION	EDUC1		
389	REGION	EDUC2		
106	REGION	ETNI		
236	REGION	PRIOCCU		
55	REGION	POSLABM		

4.2.1 Global Recoding

Via the μ–ARGUS main window the user decides which variable will be recoded. Double click on the variable in the main window or choosing the menu option Modify|Global recode brings you to the global recode window. In this window the global recodings for each variable can be specified.

Specifying and selecting global recodings



1. On the left you see a list of the variables. You can recode a variable by applying a recode scheme or by truncating a variable. Truncation is only possible if in the metadata description the variable has been designed as hierarchical.
2. If you apply a global recoding or truncate a variable, the colour of the variable will be changed into red and an 'R' or 'T ' will be indicated in the first column of the list-window.
3. If no global recode file is available you can make one yourself or read an existing one. You can also edit the recoding.
4. In the edit-box 'Codelist' you can specify the name of the file containing the codelist (labels) for the recoded variable.
5. It is also possible to change the codes for the missing values.
6. If you apply a global recoding, μ-ARGUS will show the result of the recoding in the warning-window. This can either be an error message showing on which line the error occurred, but also it can be a warning. For example some codes in the base code file have not been recoded. However this could be the purpose of the global recoding.

A recoding scheme

Global recoding means that certain categories of a variable are collapsed into one new category. The syntax of the recode file is as follows: each line in the file corresponds to one new category. The code of the new category is placed before a colon (:). The old categories to be grouped into the new category are placed behind

the colon. Single categories are separated by commas (,) and if a hyphen (-) is placed between two categories it refers to all subsequent categories between and including these two categories. If a hyphen is only placed before or after a category, this refers to all categories before or after and including this category respectively.

Example: The line "7:-4,6,8-10,13-" means that the categories 0, 1, 2, 3, 4, 6, 8, 9, 10, 13, 14, ... (if present) are recoded as 7.

Between the two windows there are five buttons.

1. "Read" allows you to read a previously made recode file into the program.
2. "Apply" applies the recode in the file to the tables. This will add the corresponding cells. If the cell was unsafe the new cell might exceed the "Safe Limit" and become safe.
3. "Truncate": Variables which are preceded by an asterisk in the left window can be truncated. This means that you are allowed to chop of digits from the end of the code for the categories. When you click on "Truncate" you can specify how many digits must be chopped. This number always applies to the original codes: if you want to truncate the same variable twice, each time one digit, you have to fill in "2" the second time.
4. "Undo" will undo any recodes, truncations and replaces performed on a variable. This brings the variable to its original state.
5. "Close" will close the dialog box and bring you back to the main window of μ-ARGUS and show the updated overview of the unsafe combinations per variable.

A further description of the global recode schemes/files can be found in section Error: Reference source not found

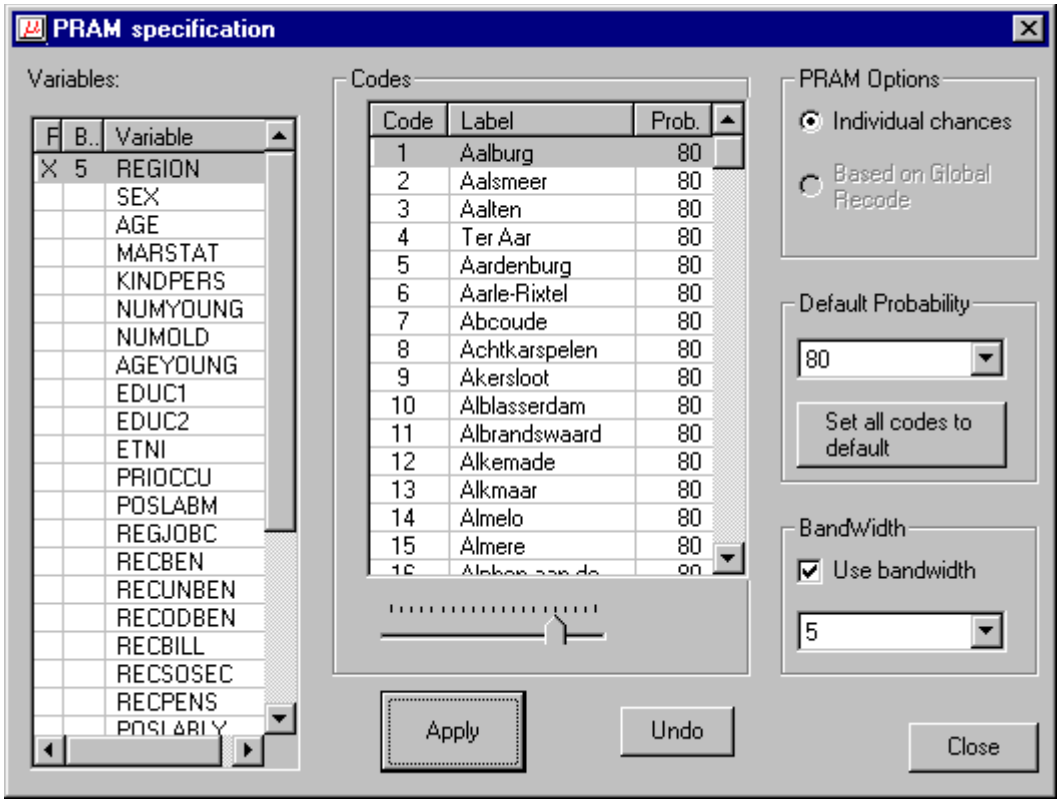
4.2.2 PRAM specification

Post Randomisation is a technique where noise is deliberately added to categorical variables. See also section 2.2.4. Additionally the parameters of the added noise can be made available to the users of a dataset. They can then make better use of the datafile. At this stage you can specify the amount of noise added to the variable. Typically this is done by specifying the chance of changing a category into an other category. In the current implementation you specify the chance of not changing a certain category. The complement is the chance of changing the value into an other category. You can limit the spread of this by specifying the bandwidth. This implies that the chance of changing a category is limited to the nearest n categories.

The individual chance per category can be changed by using the slider. Press the apply button to accept the settings.

The 'Undo'-button will reset the specification.

If a variable is PRAM-med, this is shown in the listbox by a P in the first column and an indication that the bandwidth has been used or not.



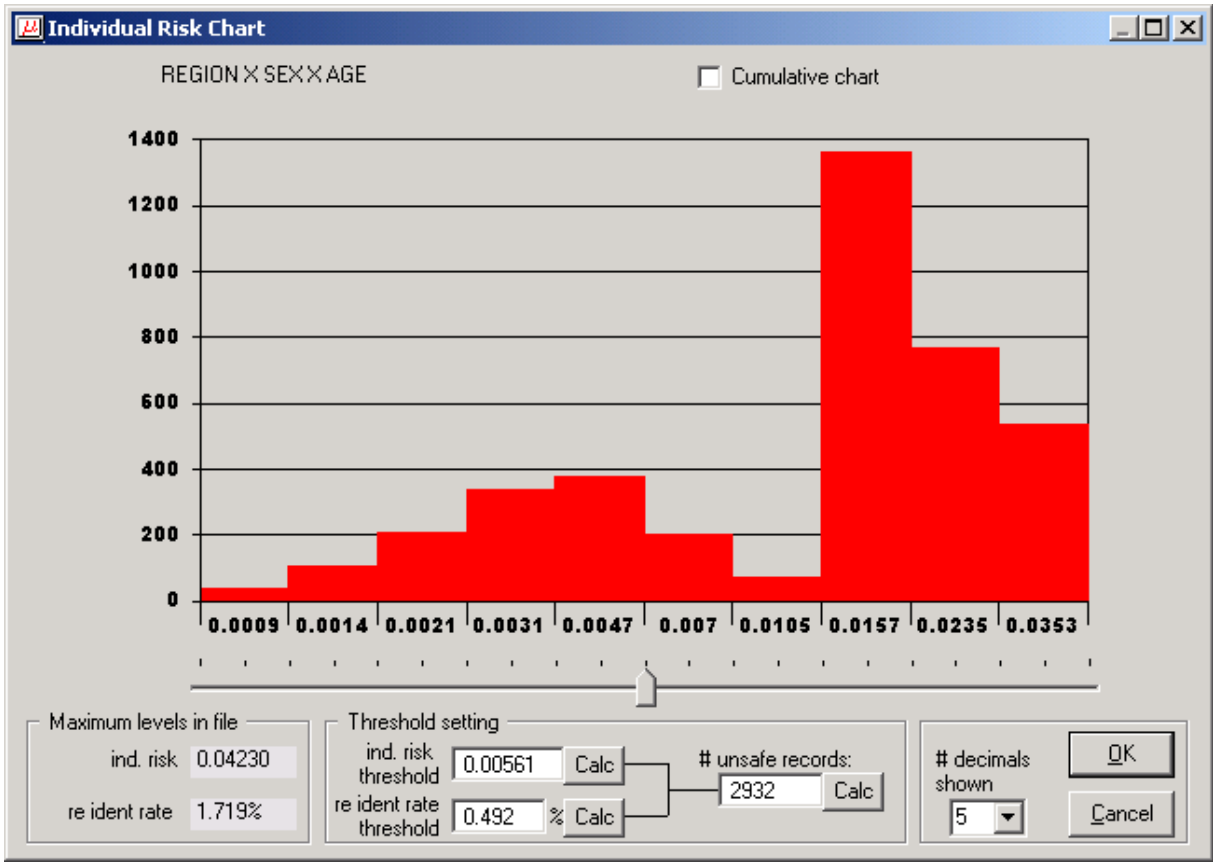
4.2.3 Risk approach

If for certain combinations the individual risk approach has been selected then you can specify the level of risk. All records with an individual risk above this level will be treated as unsafe. Local suppression will then be applied in the final stage of ARGUS. Of course it is possible to apply global recoding and re-inspect the risk chart. The level of risk acceptable for making a datafile available is of course dependent of the usage of a file, whether it goes to a research-institute or whether it is meant as a public use file.

When the graph is shown, the slider can be used to adjust the risk-level. Also the number of unsafe records is constantly adjusted as well.

If more than one risk-table had been specified, you are first asked to specify for which table you want to specify the risk-threshold.

Individual risk chart

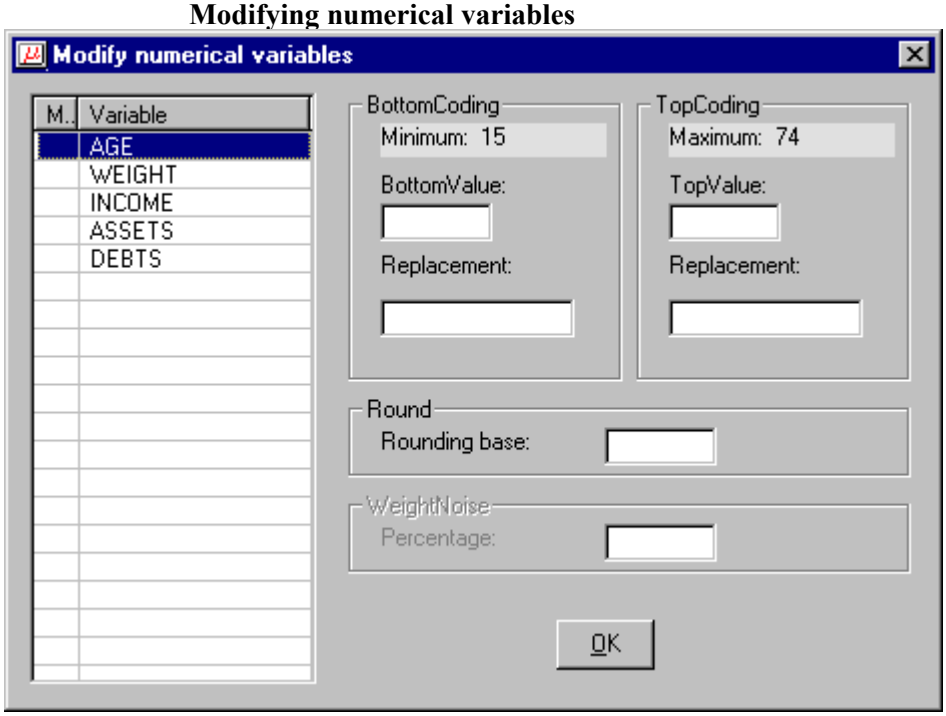


4.2.4 Numerical variables

For numerical variables you have the possibility of modifying them here. You can simply round the variable or apply top/bottom coding. Top/Bottom coding means that all values above/below a certain threshold are replaced by a given value. For rounding the rounding base should be supplied.

Additionally for the weight variable you could add noise. All these modifications will only take place when the safe datafile is generated. At this stage the information is only stored.

Modifying numerical variables

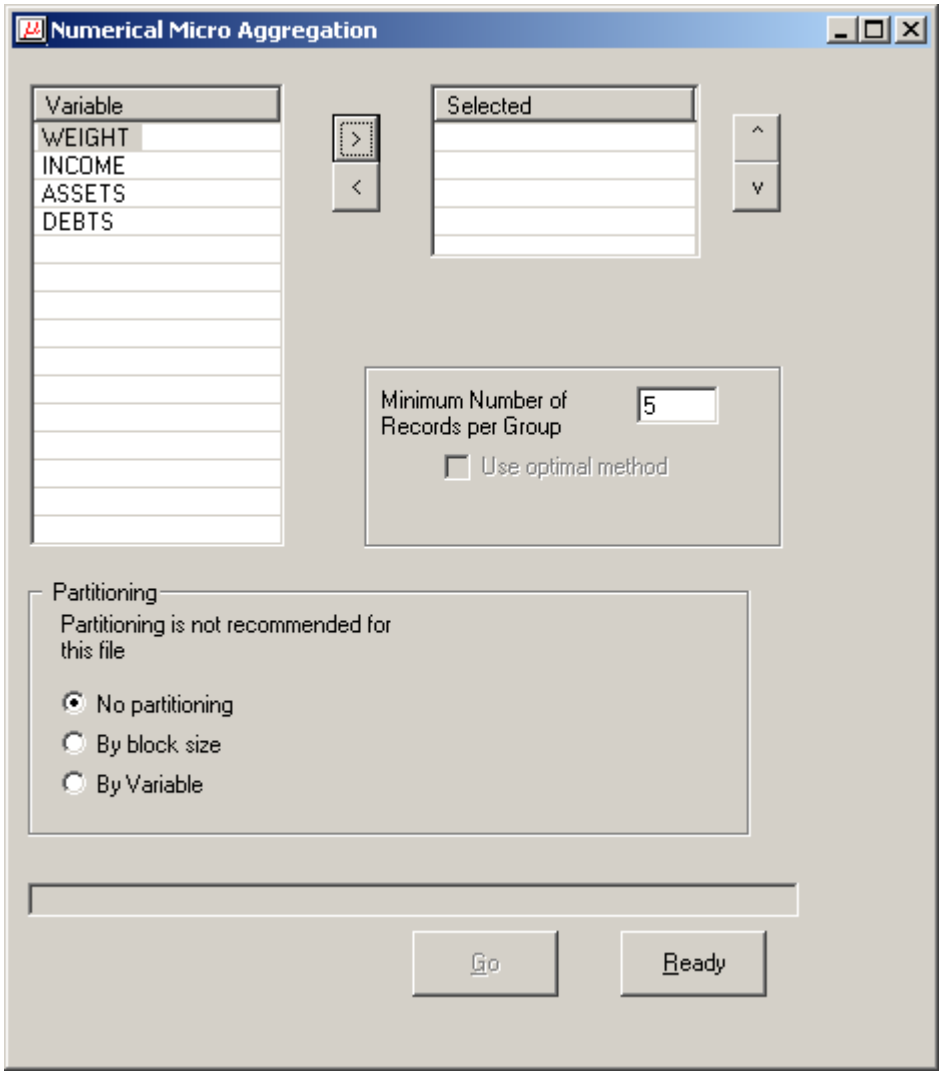


4.2.5 Numerical Micro Aggregation

To obtain microaggregates in a microdata set with *n* records, these are combined to form *g* groups of size at least *k*. For each variable, the average value over each group is computed and is used to replace each of the original averaged values. Groups are formed using a criterion of maximal similarity. Further details of the technique are given in Section Error: Reference source not found.

The user can select the variable(s) required for numerical microaggregation as well as selecting the minimum number of records per group. The user can also specify that the optimal method should be used. However, the optimal method is not available for numerical micro-aggregation of a single variable.

Numerical Micro aggregation



4.2.6 Numerical Rank Swapping

This technique, applied to variable V_i, involves first ranking values of V_i in ascending order, then each ranked value of V_i is swapped with another ranked value randomly chosen within a restricted range (*e.g.* the rank of two swapped values cannot differ by more than p% of the total number of records). However as repeated values are considered to be equivalent in rank when choosing the swap to be performed, this might lead to a few cases with a swap of more that p%. This algorithm is independently used on each variable in the original data set.

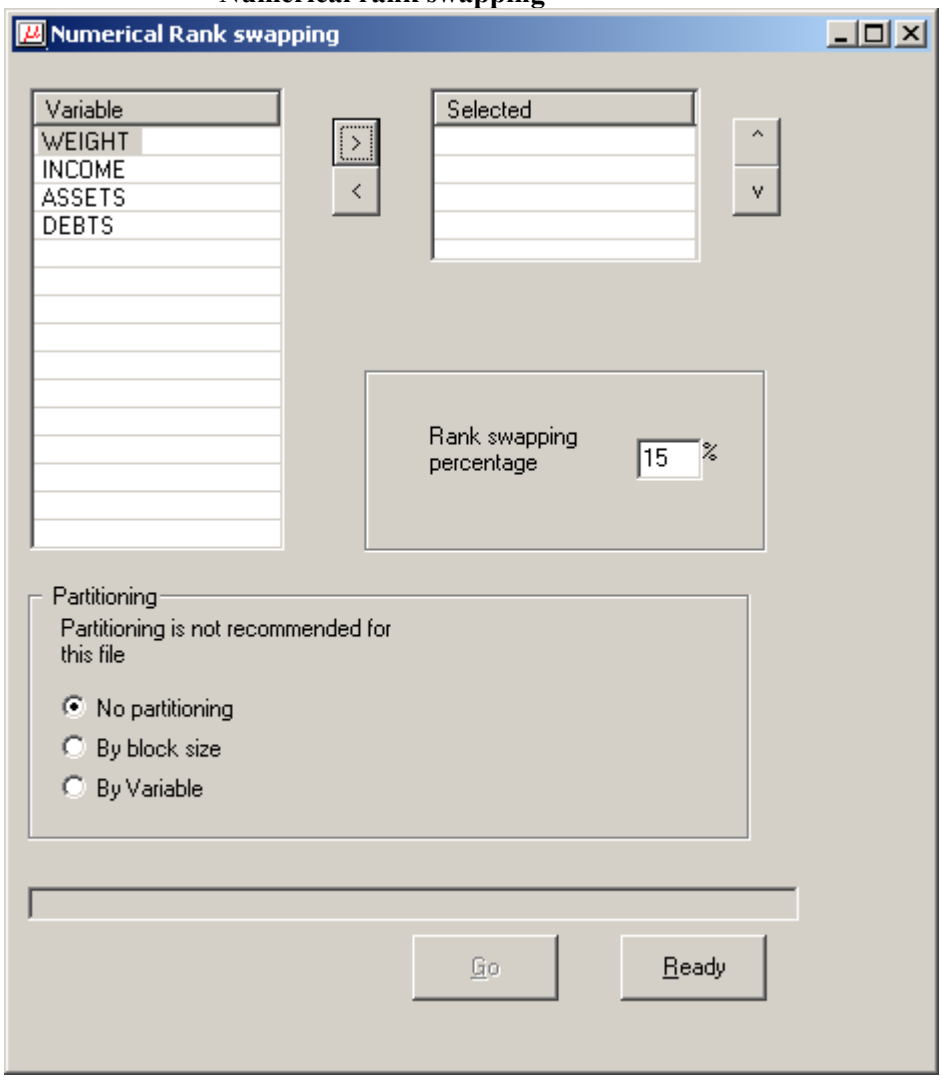
Only one parameter must be specified in order to use rank swapping:

'Rank swapping percentage': two values will not be swapped unless their rank difference is below this percentage of the total number of records. The value of this parameter must lie between 0 and 100. Values of p around 15 are recommended.

Example

If 'Rank swapping percentage' = 5, and there are 1000 rows, then a specific record will be swapped with records at a maximum rank distance of 50.

Numerical rank swapping



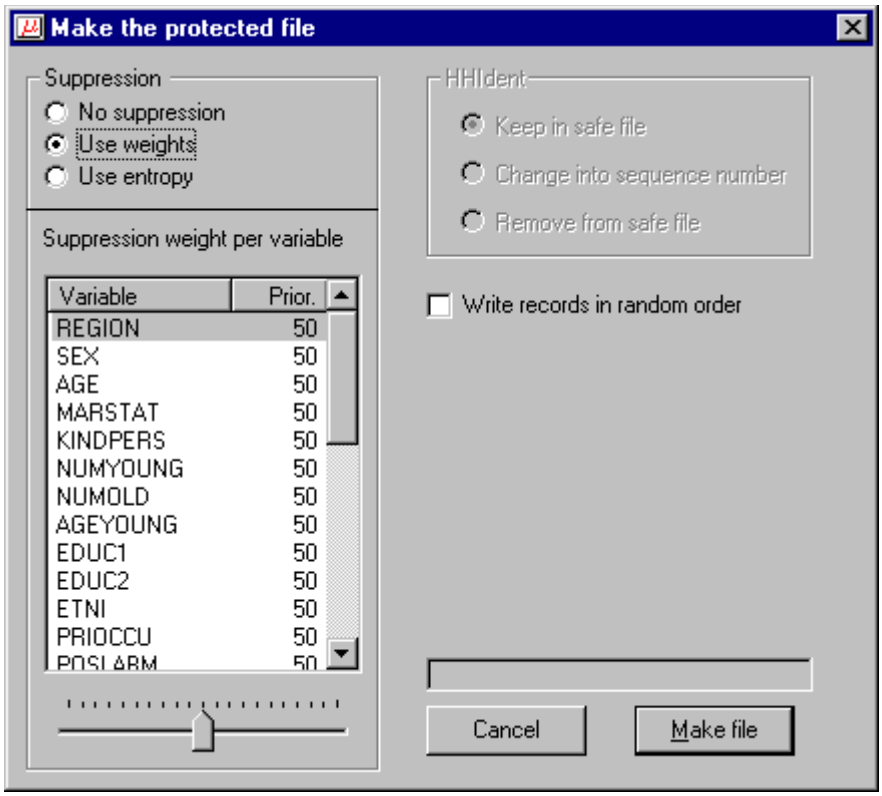
4.3 Local suppression and making a safe microdata file

When the user is satisfied with the set of global recodings, it is time to solve the remaining unsafe combinations with local suppression. For this you select the option (Output|Make suppressed file). To protect the remaining unsafe combinations μ–ARGUS will suppress certain values (i.e. impute a missing value). 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 loss of information. This information loss can be either based on the entropy function or a user-specified priority function.

The user-defined priority function can be used e.g. if a certain variable has been recoded already so much, or if a user thinks that for a certain variable the imputation of a missing value is very undesirable.

Other options are the treatment of the household identifier: keep it as it is, change it into a simple sequence number or remove it.

Also it is possible to change the order of the records in the output file (Only for fixed format micro data).



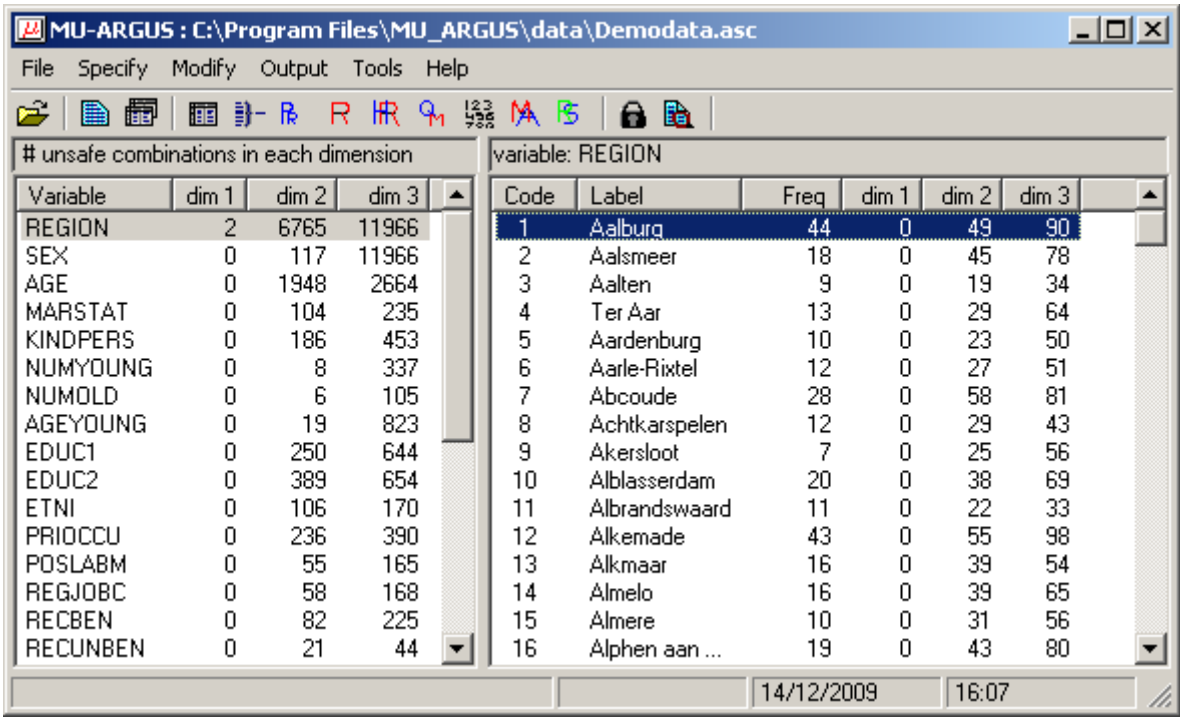
Then you press “Make File”. The program then makes new microdata, the program will apply the global recodings, replacements and truncations as ordered, and protect the last unsafe cells by locally suppressing categories in single records or sometimes, in case of a household variable, in all records belonging to the same household. If there is a weight variable it will be randomised, if requested. If there is a household identifier then there is an option to keep this identifier in the datafile, or change the household identifier into a sequence number. Please bear in mind that this variable can be very sensitive. The suggested extension of the filename is ”.saf”. It also makes a new metafile, which is the same as for the original microdata. Specific meta-information for μ-ARGUS will be removed. For the new metafile the extension is “.rds”.

When finished making the output file the program makes a report file. The format of this file is HTML. This will make it possible to view the report when ARGUS has been finished with a browser. The report file can be viewed with (Output|View report), but will also be shown automatically when ARGUS has made a safe file. This report will give a summary of the actions performed by μ-ARGUS.

5 DESCRIPTION OF MENU ITEMS

In this section we will give a description of the program by menu-item. The information in this section is the same as the information shown when the help-facility of μ–ARGUS is invoked.

5.1 Main window

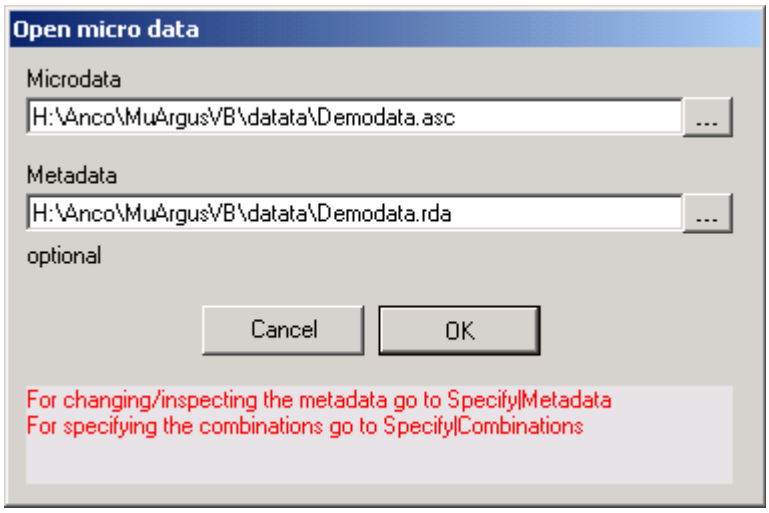


Overview of the menu-items:

File	Specify	Modify	Output	Tools	Help
Open microdata	Metadata	Show Table Collection	Make protected file	Record Linkage	Contents
Exit	Combinations	Global recode	View report		News
		PRAM specification			About
		Individual Risk specification			
		Household Risk specification			
		Synthetic data			
		Modify numeric var			
		Numerical Microagregation			
		Numerical Rank Swapping			

5.2 The File Menu

5.2.1 File|OpenMicrodata



In this dialog box you can select the microdata file and the corresponding metafile.

By default the (fixed format) microdata-file has extension .asc and the metafile .rda.

When you click on ... you get an open file dialog box. In this box you can search for the files you want to use. You can choose other file-types when you click on the files of types listbox. When you have selected the microdata-file a suggestion for the metafile is given but only when this file exists. The data file can be:

- fixed format,
- free format (with defined separator)
- 'FreeWithMeta' format. Data files in the FreeWithMeta format (with variable names specified in the headings over the columns of data) can be generated from SAS. If this input format is used, a first version of the metadata file can be generated automatically by μ–ARGUS by pressing the Generate button (in the Specify metadata window) after the data have been read into the program.
- SPSS system files. Note that it is only possible to use the SPSS system filer when a valid version of SPSS has been installed on the PC. After opening a system file it takes some time because μ-ARGUS will start SPSS in the background.

Before you click **OK** you must have filled in the name of the microdata-file.

If you change your mind you can cancel the whole operation by clicking **Cancel**.

You can specify the metadata file here as well. If all metadata is ready you can go directly to the Specify Combinations. Otherwise you will have to specify the layout of the data first via Specify metadata.

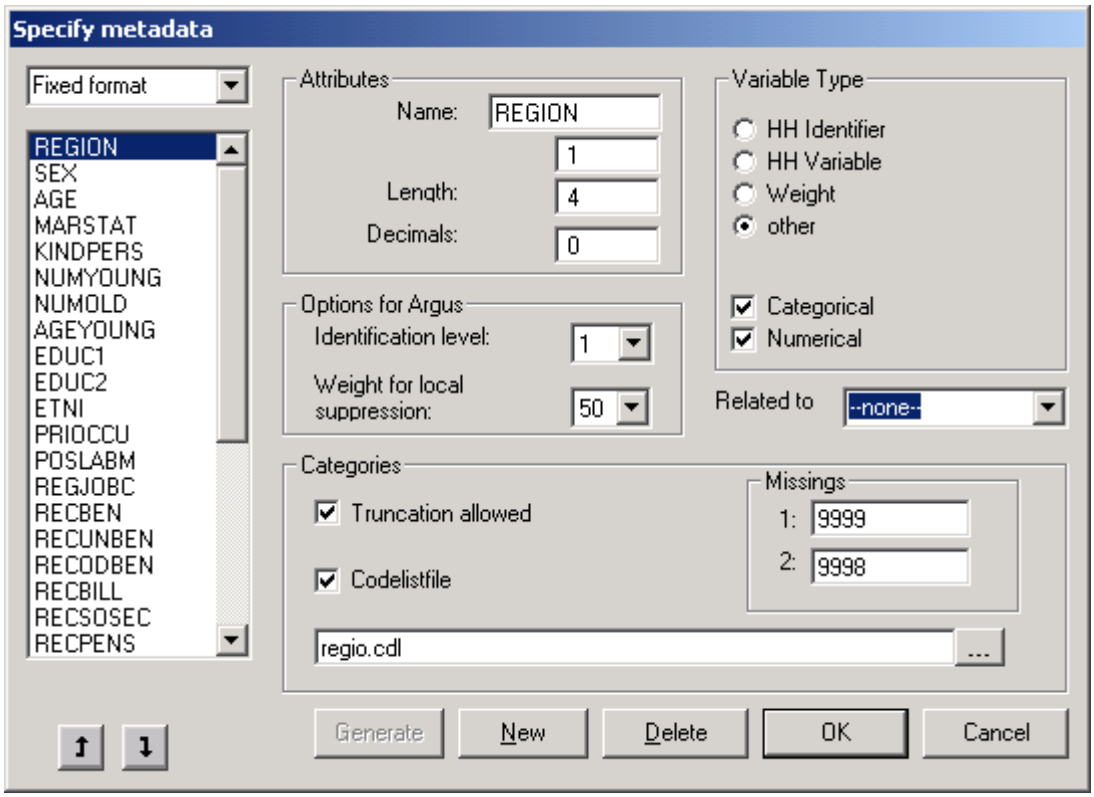
If your data is stored in a SPSS systemfile most of the metadata is copied from the SPSS system file when selecting the variables, while you can add the SDC-specific metadata after that. The complete metadata can be stored in a RDA-file later.

5.2.2 File|Exit

Terminates the μ–ARGUS-session.

5.3 The Specify menu

5.3.1 Specify|MetaFile



In this dialog box all the attributes of the identifying variables can be specified.

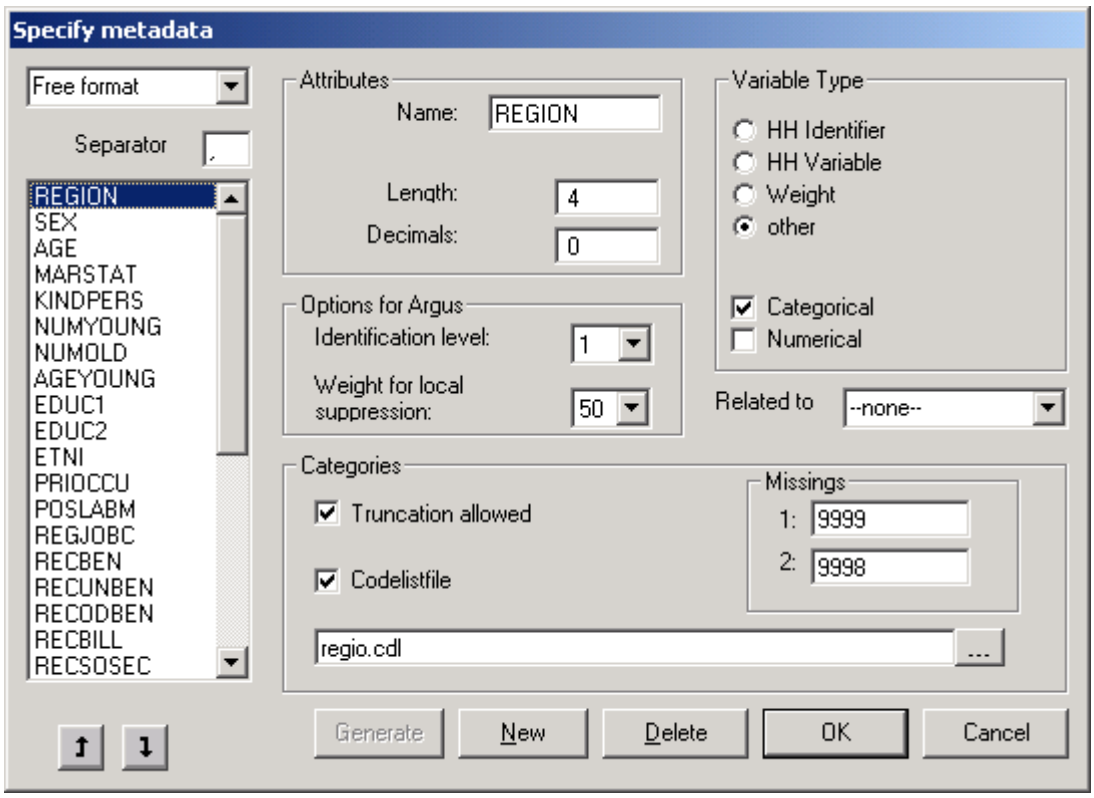
Attributes

If no rda file has been specified the information can be specified in this dialog box after pushing the New button. As a default "new" is substituted. Apart from defining a new variable, an existing one can be modified or deleted.

Fixed format data file

The following attributes can be specified: name of the variable, its first position in the data file, its length and the number of decimal places. Furthermore the kind of variable can be specified: weight variable, household variable, household identifier, or none of these. A weight variable specifies the weight of the record, and is based on the (post)sampling design used. A household variable is one that yields the same score for individuals belonging to the same household. A household identifier uniquely identifies households represented in the file.

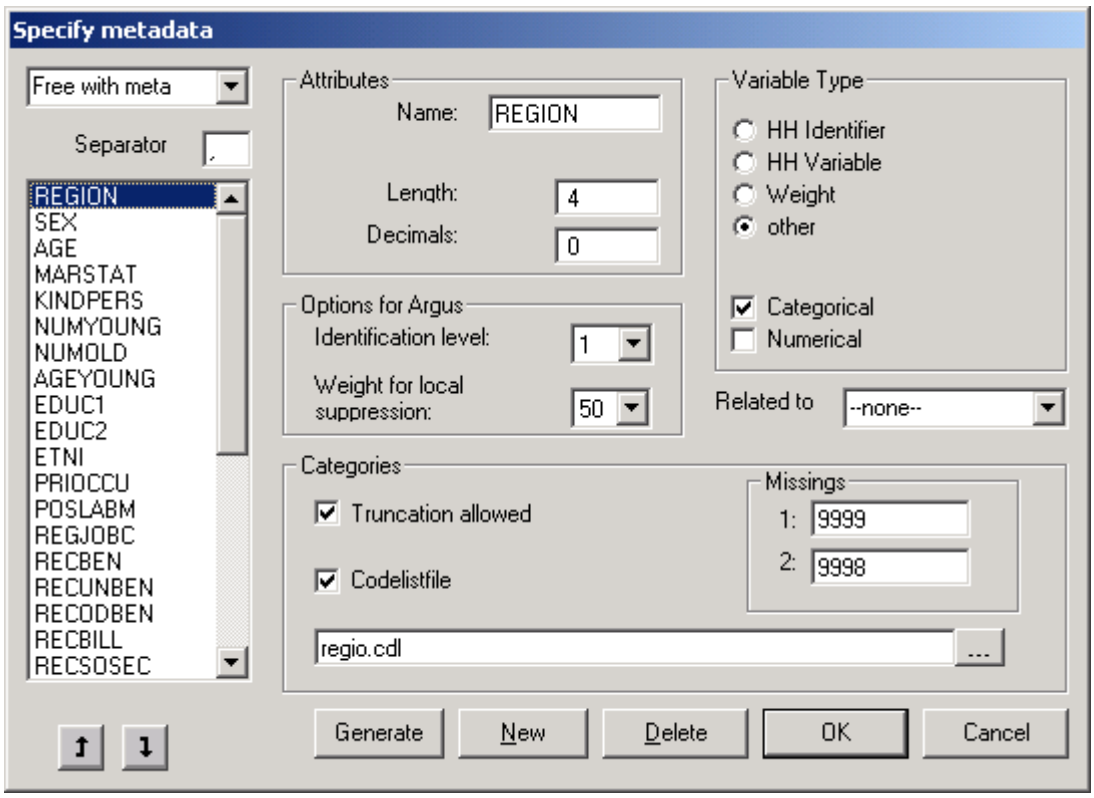
Free format data file



The following attributes can be specified: name of the variable, the character used as a separator between different data items, the length of the variable, and the number of decimal places. Furthermore the kind of variable can be specified: weight variable, household variable, household identifier, or none of these. A weight variable specifies the weight of the record, and is based on the (post)sampling design used. A household variable is one that yields the same score for individuals belonging to the same household. A household identifier uniquely identifies households represented in the file.

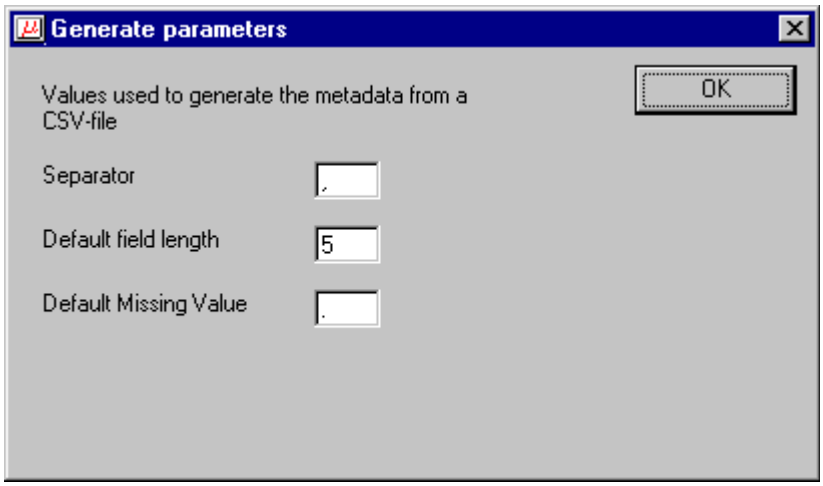
The ‘Up’ and ‘Down’ buttons can be used to change the order of the variables. Different from the fixed format record description the order of specification is necessarily the order in which the variables are stored in the free-format datafile.

FreeWithMeta format datafile



The free format with meta format is a special variant of a free format data file. E,g, SAS users can easily generate this format. The difference with the normal free format datafile is that the first record contains the names of all the variables.

The following attributes can be specified: name of the variable, the character used as a separator between different data items, the length of the variable, and the number of decimal places. Furthermore the kind of variable can be specified: weight variable, household variable, household identifier, or none of these. A weight variable specifies the weight of the record, and is based on the (post)sampling design used. A household variable is one that yields the same score for individuals belonging to the same household. A household identifier uniquely identifies households represented in the file.



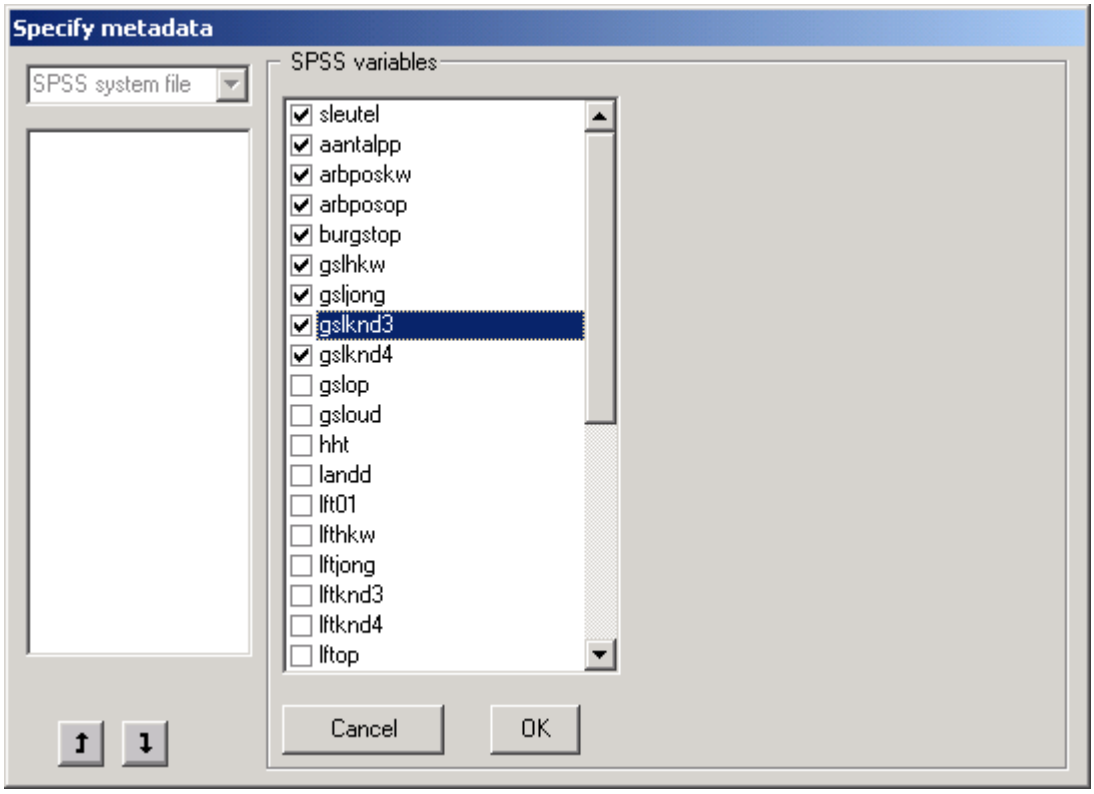
If this input format is used, a first version of the metadata file can be generated automatically by μ-ARGUS by pressing the Generate button (in the Specify metadata window) after the data have been read into the program.

SPSS system file

When using an SPSS systemfile for the fist time you do not have to specify a metadata (RDA) file. Most of the metadata is retrieved from SPSS. When opening the SPSS system file for the first time, leave the name of the RDA file open. In the Specify|Metadata window press the new button. This will open the following window

Selecting variables form an SPSS system file

:



You only need to select those variables that will play a role in the μ-ARGUS session. After pressing the OK button you will get the familiar Specify|Metadata window. You will then have to add the SDC-specific metadata. After that you are ready for a normal μ-ARGUS session. Note that when you will have protected the

data file, μ-ARGUS will automatically update the SPSS system file when you ask to save the data.

SDC-options for ARGUS

The identification level of the variable can also be specified. This level is used to let μ–ARGUS generate combinations of variables that should be checked. Identification level 0 means that the variable is not identifying, implying that it does not appear in any of the combinations to be checked. Identification level 1 is the highest level of identifiability, followed by levels 2, 3, etc. (Identification levels ≥ 4 leads to very large sets of combinations to be checked). If a variable is identifiable at level i, it is also identifiable at level i+1. In other words the identification levels (for i>0) are nested.

Also the suppression priority (weight for local suppression) can be specified here already. It will play a role when the safe datafile is created at the end of a μ-ARGUS-run.

Codelist

The codelist of a variable should be known to μ–ARGUS before it can operate properly. There are two options for obtaining the codelist for each variable. In the first place they can be specified by the user, by listing them in a codelist filename. Or such a codelist (or rather the part that appears in the data) can be determined by μ–ARGUS by a run through the microdata.

Hierarchical variables can be truncated, that is, some digits of the codes can be "chopped off" and then yield a meaningful code. In this way one can replace a ‘n+1’ digit code by a ‘n’ digit code, by chopping off the least significant (n+1th) digit. A variable can have two missing values. The first one is the one that is used by μ–ARGUS when it locally suppresses a value of this variable. The second missing value is only used by μ–ARGUS to know that this value should not be used as a regular, i.e. nonmissing, value. In many surveys two missing values are used for don't know and refusal.

Related to

At the end of a μ-ARGUS session μ-ARGUS will suppress certain remaining unsafe combinations. Sometimes variables have a strong relation between each other. So suppressing variable 1 might be useless if not also variable 2 (related to 1) will be suppressed. If the relation is indicated here μ-ARGUS will take this into account.

The meta data is stored in an RDA-file. A further description of this file can be found in section 6.1

5.3.2 Specify|Combinations

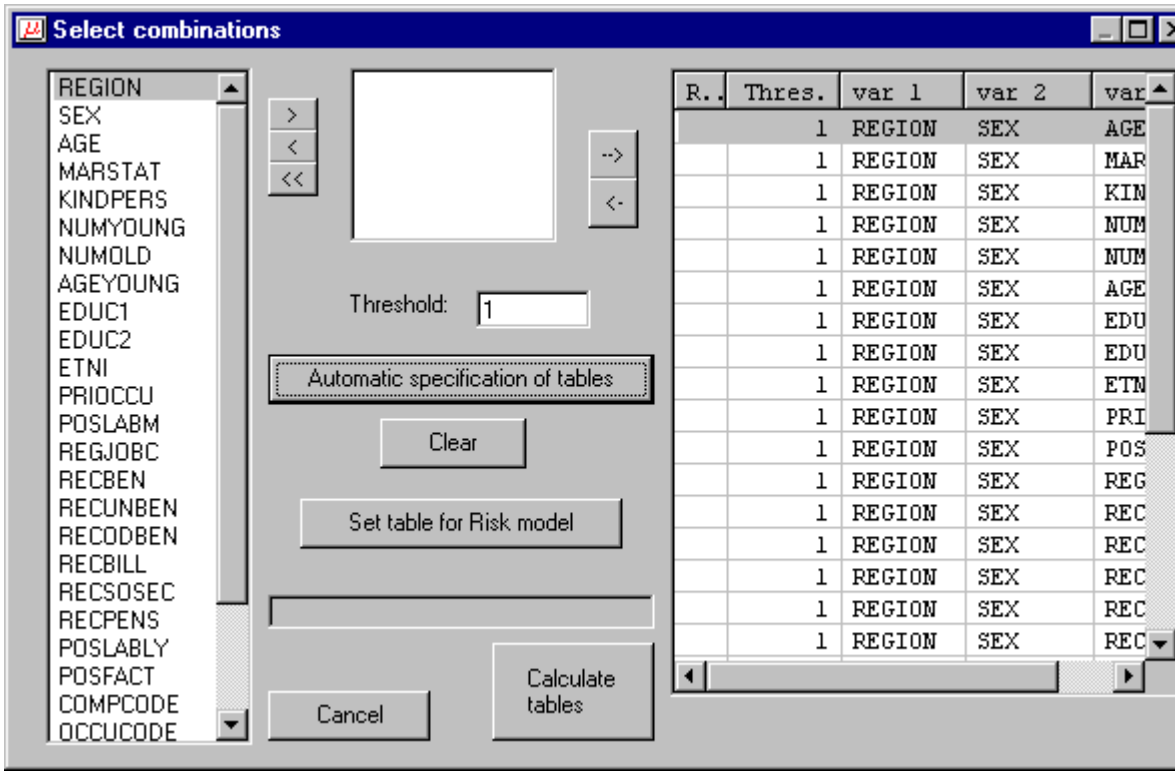
This dialog box can be used to specify the tables that should be checked. This option can be used in one of two ways. First of all a user can interactively select variables from the listbox that should span a table. Secondly it can be used to let μ–ARGUS generate the tables. The package can do that in one of two ways:

- by generating all combinations of k variables. Only the variables with identification level >0 are used. K is to be specified by the user. For each dimension the threshold value should be specified. This option is the implementation of the Dutch approach for Public Use Files (PUF).
- by using the identification levels of all variables, provided that they have been specified by the user. In this case μ–ARGUS generates all tables that can be obtained when variables from different levels of identification are chosen. In this case a single threshold should be specified. This option is the Dutch approach for Micro data files Under Contract. (MUC).

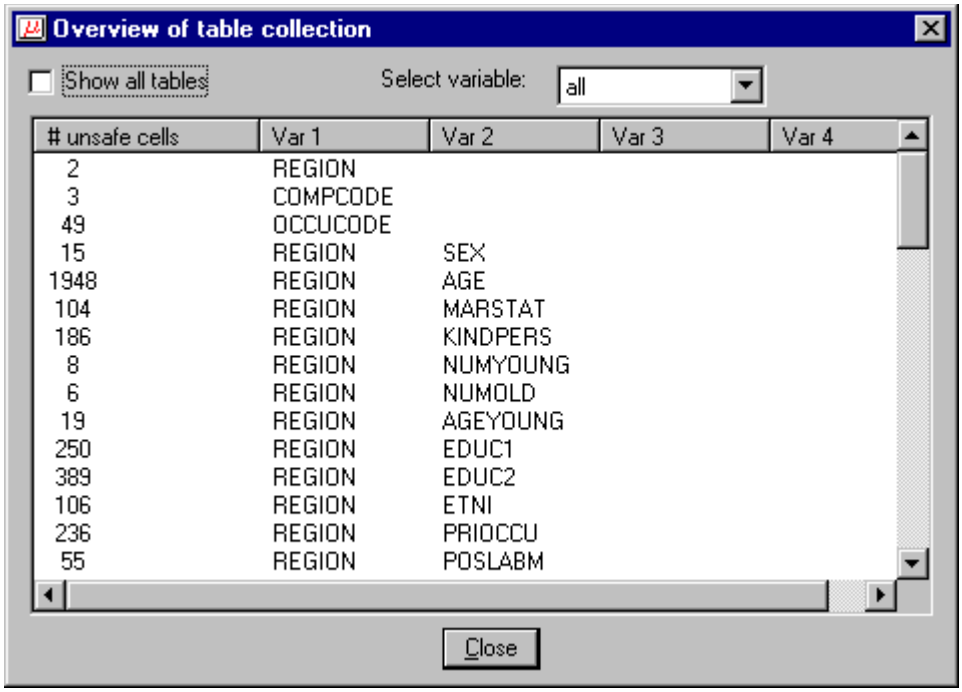
There are undo options that allows one to remove variables from a table or to remove tables that have been selected at an earlier stage.

Additionally there is the option of specifying that a table will be used for the new Individual risk approach. Because the traditional risk model and the new apprach cannot be applied to the same identifying variables, all overlapping combinations are removed,when a table is used for the individual risk model.

The window for the specification of the combinations



5.4.1 Modify|Show Tables

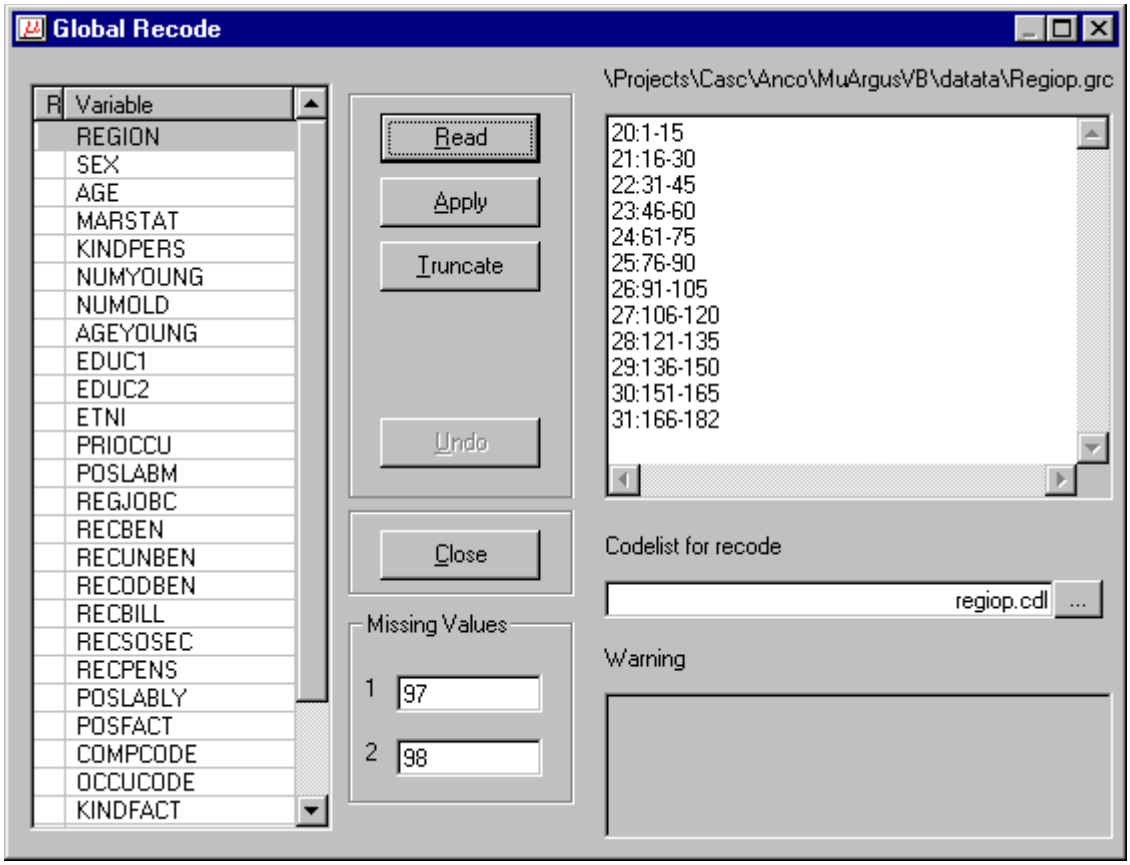


In this window you can see which tables have been generated and how many unsafe cells each table has.

- On default only the tables with unsafe cells are shown. You can see all tables when you check **Show all tables**.
- If you are only interested in one variable you could use the select variable box
- The **Close** button closes the dialog box.

5.4.2 Global Recode

Global recoding (in conjunction with local suppression) is the main option for protecting individual information in a data file.



This dialog box offers two possibilities to reduce the number of unsafe cells. These are global recode and truncate.

Short description

- The left pane shows a list of all the variables that can be recoded.
- If it is possible to truncate a variable the truncate button is active.
- With **Read** you can open existing recode files.
- If you click on the right edit box, you can easily change a recoding scheme
- With **Apply** the recoding scheme will be applied to the selected variable.
- With **Truncate** the truncation will be applied to the selected variable.
- All recoding and truncations are applied on the original categories. When a recoding, truncation or replacement is applied, a **R** or **T** is placed in front of the variable, and the variable is shown in red.
- With **Undo** you can undo a global recoding or truncation of the variable which has been selected.
- The **Close** button closes the dialog box.

Global Recode

There are some rules on how you have to specify a recode scheme. All the codelists are treated as alphanumeric codes. This means that the codelist are not restricted to numbers only. However this implies that the codes '01' and ' 1' are considered different and also 'aaa' and 'AAA' are different. In a recoding scheme you can specify individual codes separated by a comma (,) or ranges of codes separated by

a hyphen (-). Special attention should be paid when a range is given without a left or right value. This means every code less or greater than the given code. In the first example below the new category 1 will contain all the codes less than or equal to 49 and code 4 will contain everything larger than or equal to 150.

Example:

For a variable with the categories 1 till 182 a possible recode is then:

1: - 49
2: 50 - 99
3: 100 - 149
4: 150 -

For a variable with the categories 01 till 10 a possible recode is:

1: 01 , 02
2: 03 , 04
3: 05 - 07
4: 08 , 09 , 10

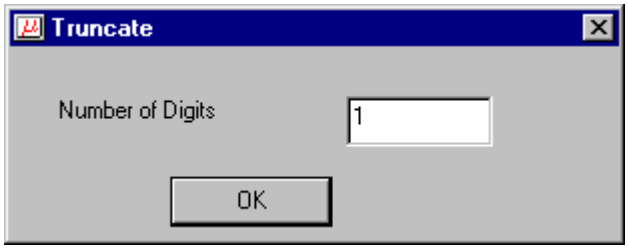
Don't forget the colon (:) if you forget it the recode will not work.

The recode 3: 05-07 is the same as 3: 05,06,07 you can choose what you like best.

A further description of the global recode file can be found in section Error: Reference source not found

Truncate

If you click **Truncate** this dialog box will pop up.



You can specify how many digits you want to truncate. If you apply a truncation and you want to truncate another digit you have to fill in 2 digits in the popup dialog box. This is because the recodings are always applied to the original categories.

Codelist

A new codelist can be selected for the recoded variable. Note that similar to the codelists in the meta data file (.RDA), this codelist is only used for enhancing the information reported on a variable in the main-window of μ-ARGUS.

Missing values

When a new recoding scheme is selected also a new set of missing values can be specified. If these boxes are empty the original missing values will still be used.

Warning

In the warning pane μ-ARGUS reports back on a recoding that has been applied. Sometimes these are only warnings, e.g. only a part of the codelist has been recoded. Sometimes a severe error has occurred, preventing μ-ARGUS from applying a recoding scheme. Typically the syntax of a recoding scheme has been violated.

When the focus is moved to another variable, or when this window is closed, μ-ARGUS will ask you whether the recoding scheme must be saved. Note that also the changed missing codes and the name of the codelist are stored in this file. A typical extension is .GRC.

Example of a .GRC-file

```
   20:1-15
   21:16-30
   22:31-45
   23:46-60
   24:61-75
   25:76-90
   26:91-105
   27:106-120
   28:121-135
   29:136-150
   30:151-165
   31:166-182

   <MISSING> 97 98
   <CODELIST>
regiop.cdl
```

5.4.3 PRAM specification

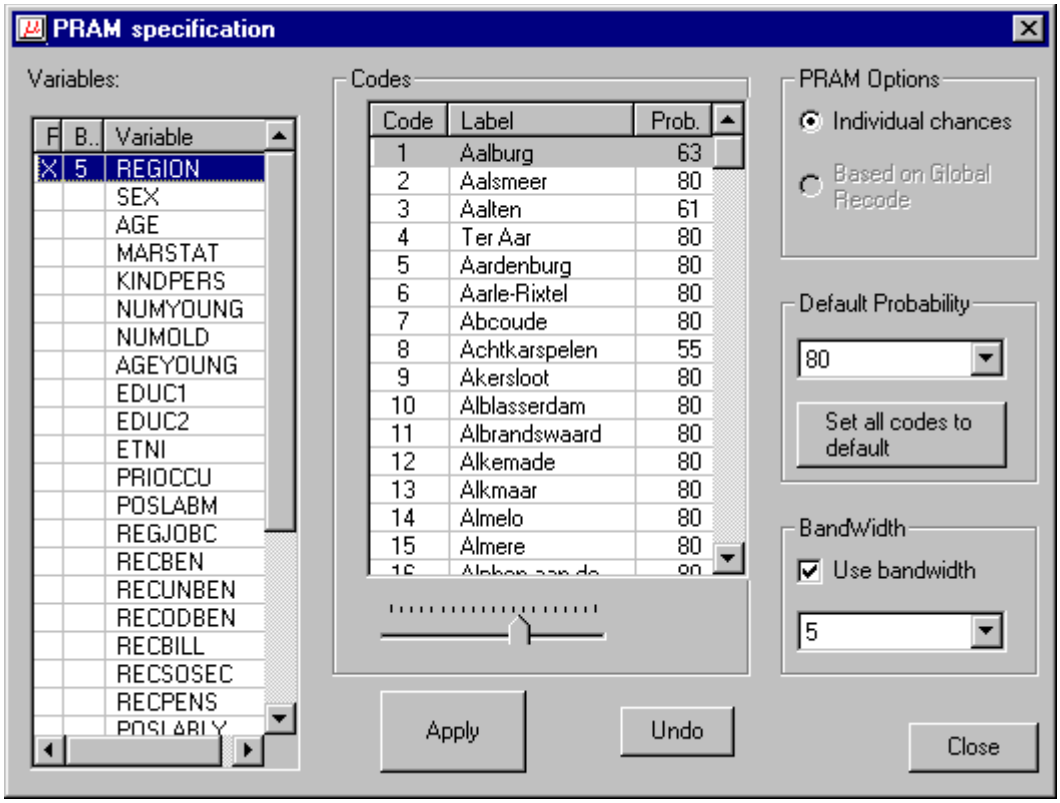
PRAM is a disclosure control technique that can be applied to categorical data. Basically, it is a form of deliberate misclassification, using a known probability mechanism. Applying PRAM means that for each record in a microdatafile, the score on one or more categorical variables is changed. See also section 2.2.4.

At this stage you can specify the probability of changing the score of a category.

First you select a variable to be PRAMmed. In the listbox in the middle you will see all the categories. The probability is the probability of not changing a score. You can set all probabilities to a certain percentage with the button 'Set all codes to default'. Further it is possible to change individual scores using the slider.

If a category will be changed, randomly another category will be selected. However if you want to restrict this, you can select a bandwidth. In that case the new category will be a neighbouring one.

Pressing the 'Apply'-button will store the information. The actual PRAMming will only be done when the new datafile is generated. It is still possible to apply global recoding and come back to re-specify your PRAM-probabilities.

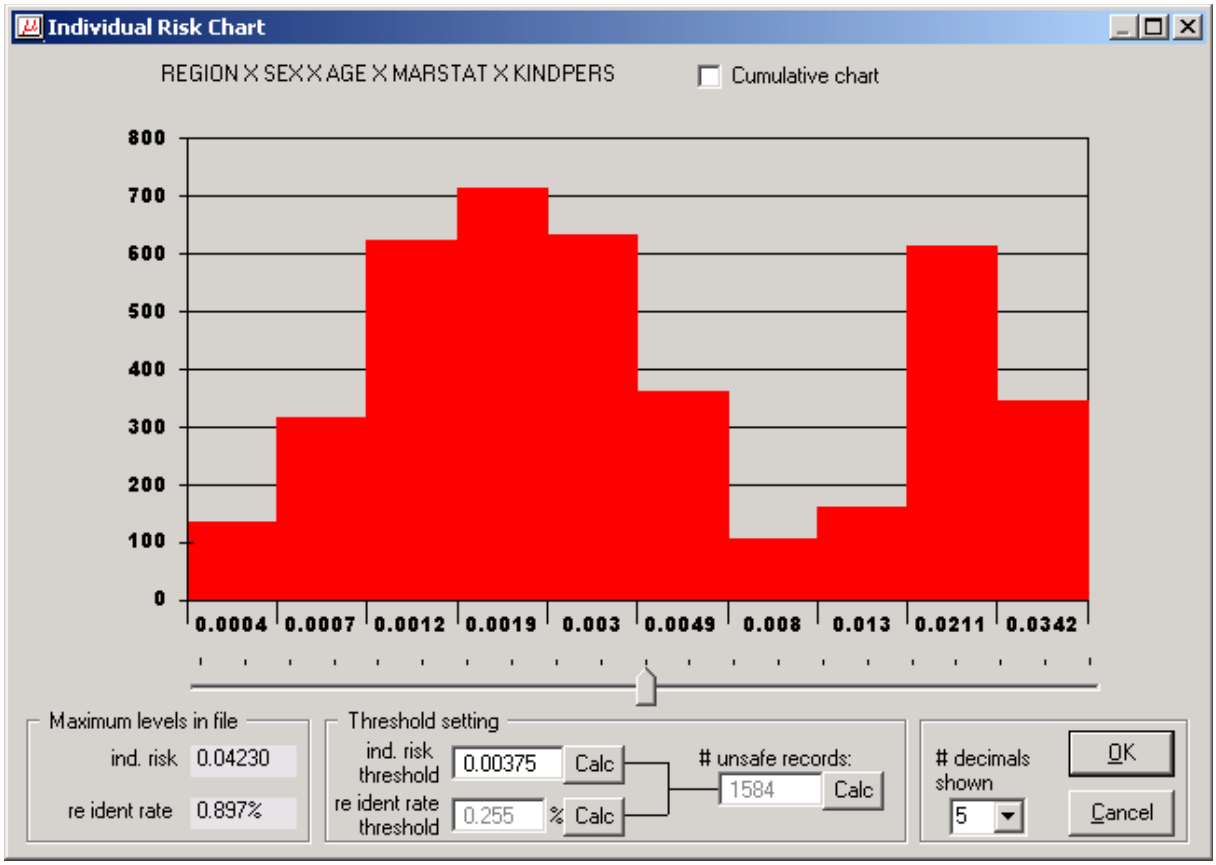


5.4.4 Individual risk specification

A new risk model has been incorporated in μ-ARGUS. This method is based on the work by Luisa Franconi and her colleagues at Istat. A full description can be found in section 2.3 and 2.4. Here you can specify the threshold level for the risk-model. All records above the chosen risk-level are considered unsafe. When generating a safe file in the final stage of μ-ARGUS the combination of variables on which the risk model is based is considered unsafe and a missing value will be imputed.

Use the slider to set a risk-level. You will see at the same time the number of records that will be considered unsafe, when you fix this risk level.

When you have fixed a risk-level, you can still go back, select an other global recoding and inspect the risk-chart again.



5.4.5 Synthetic Data

Warning/Installation: Before this option can be used, an additional installation procedure must be followed. The synthetic data procedure was written in by Ursula Gonzales and Josep Domingo. The procedure was written in R. Therefore a valid installation of R is required. R is an Open Source package for statistical analysis and can be downloaded free. And the procedure must be loaded in R.

Three packages are needed:

Som

ConCor

HybridIPSO

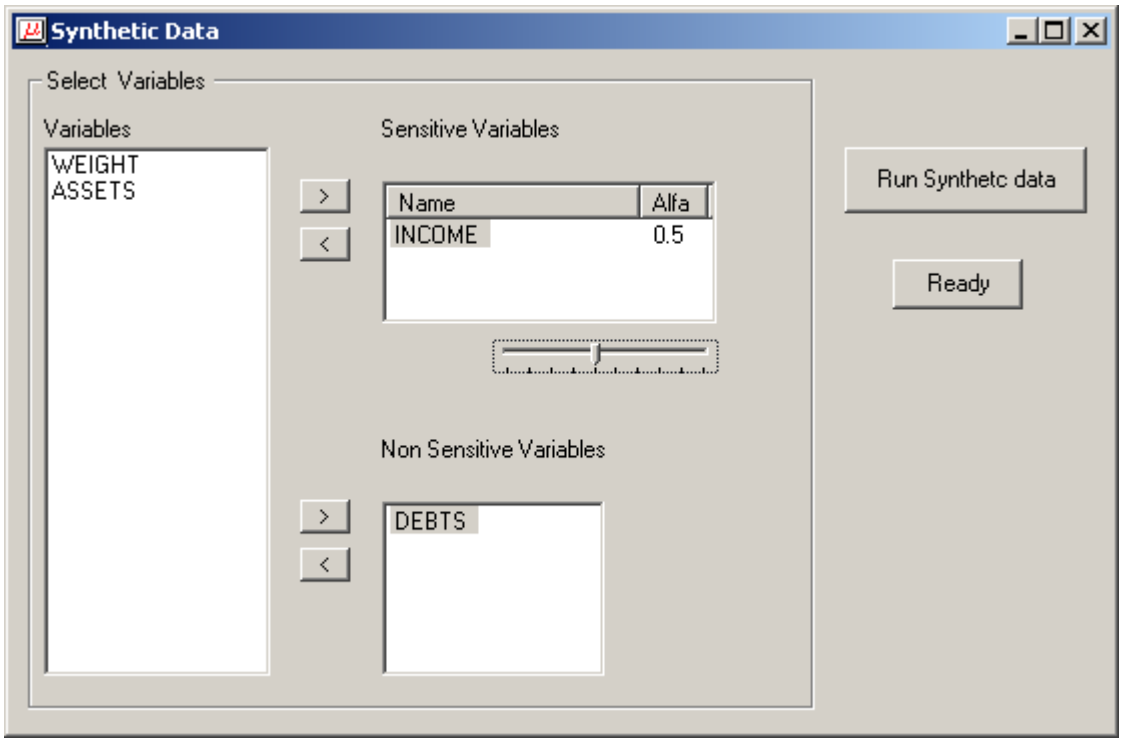
After the installation of μ-ARGUS these three packages can be found in a sub-directory of μ-ARGUS. Usually the μ-ARGUS-directory will be in C:\Program Files/MU_ARGUS.

Maybe Som and ConCor have been loaded for other R-applications. They are also available from the standard R-website. If they have been loaded already, you only have to load HybridIPSO. Use the procedure Packages|Instal package(s) for local zip-files from the main menu of R.

This all has to be done only once.

Finally you will have to tell μ-ARGUS where he can find R. The first time you will use the synthetic data procedure, you will be asked to indicate where R, more precisely the R executable (R.exe) can be found.

A file open box will be displayed. A standard location will be something like: C:\Program Files\R\R-2.9.0\bin\r.exe, but your local installation can be different. μ-ARGUS needs to know the location of the binary file R.EXE. μ-ARGUS will remember this information and not ask it again.



Performing the generation of synthethic data

In the above window the variables selected for generating synthetic data can be specified.

The sensitive variables can be selected to the first listbox and the non-sensitive variables to the second list-box. Also the alpha-parameters (see section 3.2.8) can be specified by selecting the variable and changing the slider. Note that a value < 1 and > 0 must be chosen. As the procedure only works with numerical data, only the variables declared numerical in the meta data can be used for the synthetic data generator.

Pressing the button 'Run Synthetic Data' will start the computations. Input files for the R-procedure will be prepared, the R-system will be started and finally the results will be stored.

5.4.6 Modify numerical variables

Several modifications to a numerical variable are possible.

- Top/Bottom coding

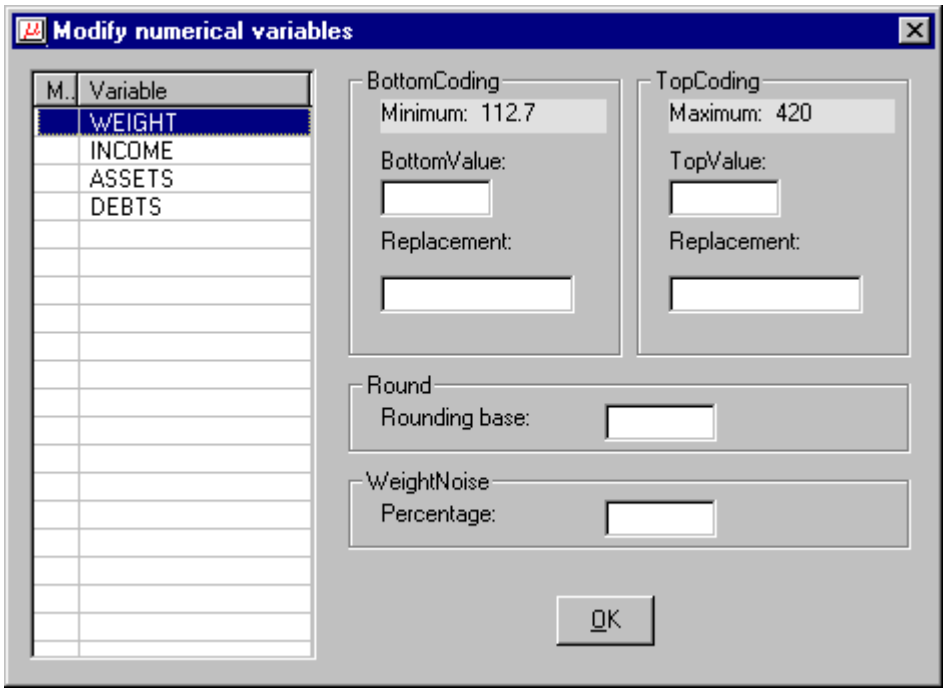
- Rounding
- Add noise to the weight variable

Top and bottom coding is a technique to protect the extreme values of a distribution. Typically it is applied to numerical variables. All values above respectively below a certain threshold value will be replaced by another value. In this stage of using μ–ARGUS only the instructions will be stored. The actual top/bottom coding will be done when the safe microdata file is written.

In this window μ-ARGUS shows the real minimum and maximum of a variable. You can specify a value above/below which all values are replaced by a specified replacement value. μ-ARGUS is completely flexible what replacement you specify. The replacement value can also be alphanumeric. μ-ARGUS only prevents you from entering a top/bottom value above/below the extreme value.

Rounding can be applied to a numerical variable. It will increase the protection of the datafile, however there are no real measures to calculate the increase in protection.

Weight/Noise. As the weight variable might in some cases be identifying, it might e.g. disclose the municipality, adding noise can be a solution. At this stage you can specify the amount of noise to be added as a percentage.



5.4.7 Numerical Micro Aggregation

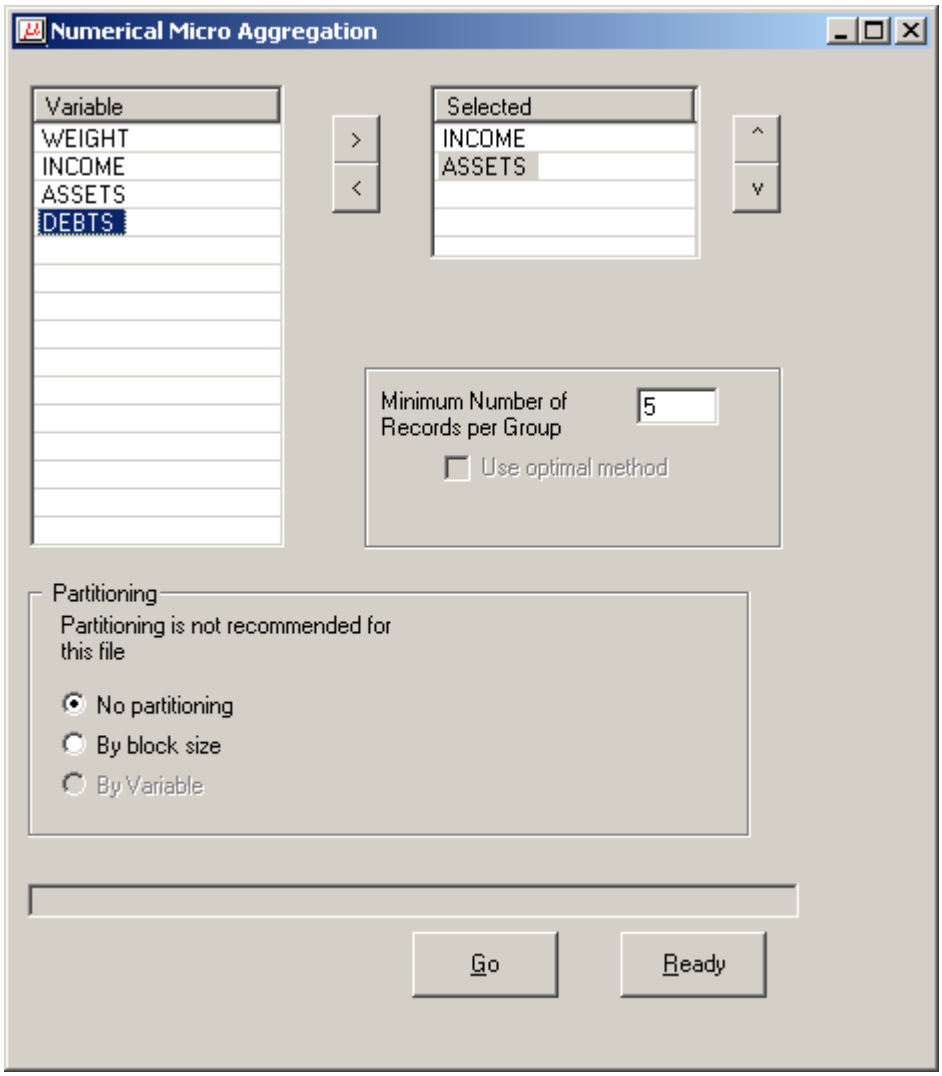
Numerical Micro Aggregation can be applied to numerical variables only. The user can specify whether the optimal method should be used or not. However, the optimal method is not available for numerical micro-aggregation of a single variable.

The minimum number of records per group can also be specified

To run the program click on the 'Go' button. The 'Ready' button will take the user back to the main menu.

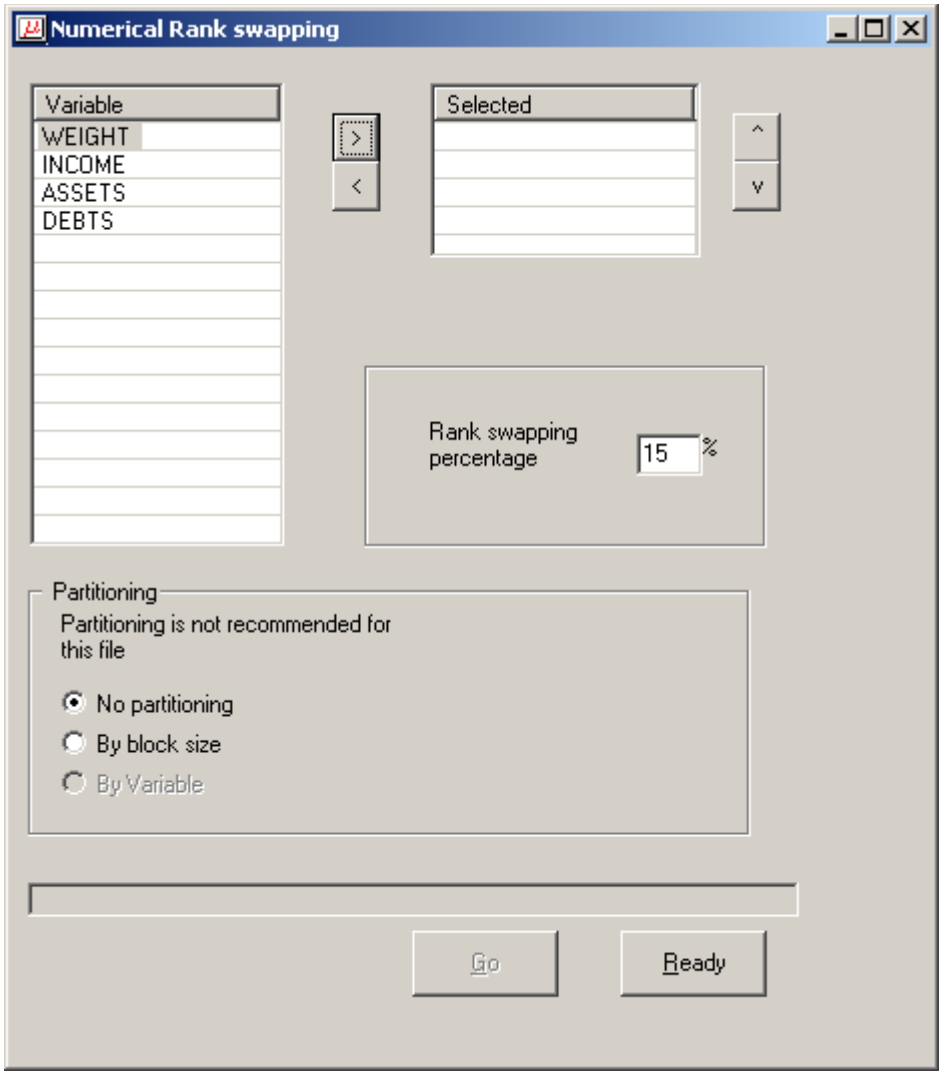
For very large data files the capacity of the computer might become a problem. Although an efficient algorithm is used, partitioning the datafile into smaller pieces might be a good solution. This partitioning (also called blocking) can be done by cutting the file into subfiles by choosing the number of blocks.

Alternatively a blocking variable can be selected. E.g. a regional variable can be used here very well. Applying the micro aggregation procedure by province could be a very sensible solution.



5.4.8 Numerical Rank Swapping

The user choices here are to select the numerical variables required for rank swapping and choose the rank swapping percentage. If 5% is selected and there are 1000 rows, then a specific record will be swapped with records at a maximum rank distance of 50.



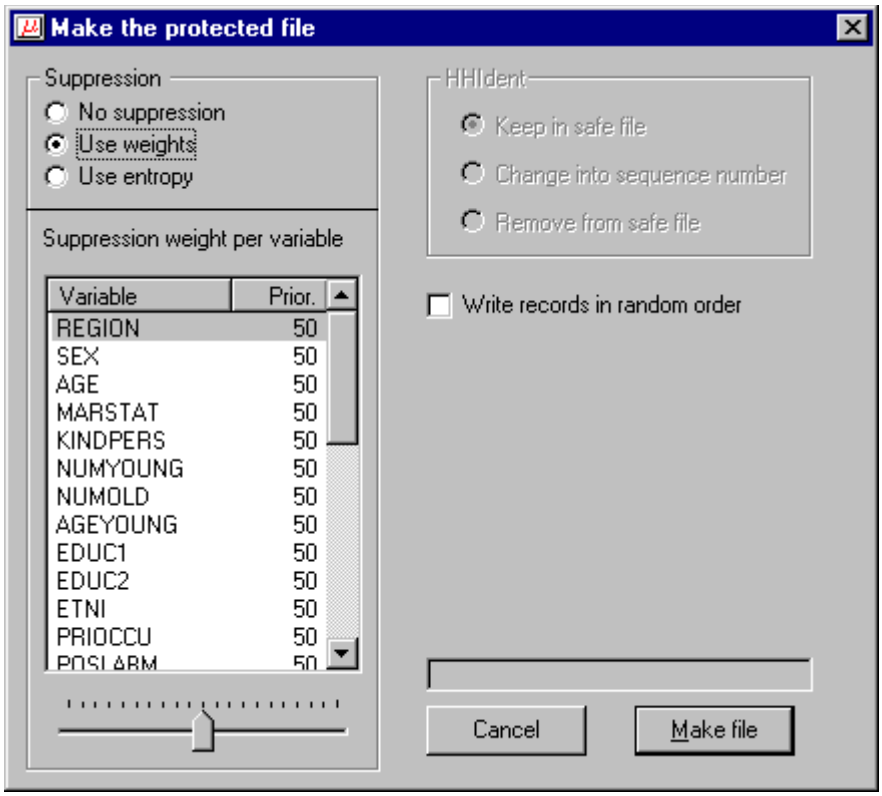
The 'Go' Button performs the operation and the 'Ready' button takes the user back to the main menu.

For very large data files the capacity of the computer might become a problem. Although an efficient algorithm is used, partitioning the datafile into smaller pieces might be a good solution. This partitioning (also called blocking) can be done by cutting the file into subfiles by choosing the number of blocks.

Alternatively a blocking variable can be selected. E.g. a regional variable can be used here very well. Applying the micro aggregation procedure by province could be a very sensible solution.

5.5.1 Output|Make Suppressed File

When finally all the data modifications have been specified, Global recoding, Risk specification, PRAM-parameters, Top/bottom coding, rounding, noise added to the weight variable, it is time to write a protected file.



When actually writing the safe file, the data manipulations are applied and the remaining unsafe combinations are protected by local suppression. I.e. certain values are replaced by the first missing value. Which variable(s) are selected is a small optimisation problem. If there is only one unsafe combination the variable with the lowest information loss is chosen. To calculate the information loss there are two options. Either you select the 'use weights' and you are free to assign an information loss (suppression weight) to each variable. The variable with the lowest information loss is then suppressed. The alternative option is to use an entropy function. The variable with the lowest value of the entropy function will then be suppressed. This entropy H(x) is defined as

$$H(x) = -\frac{1}{N}\sum_{x \in X} f(x)\log_2 \frac{f(x)}{N}$$

where f(x) is the frequency of category x of variable X and N the total number of records.

In the case of more than one unsafe combination in a record a set of variables must be suppressed. This set is chosen in such a way that all unsafe combinations are protected, but the total information loss is being minimised. Note that if a

household variable need to be suppressed in a record it is suppressed in all records of the same household.

If there is a household identifier, there are three possibilities

- Do not change
- Change the Household identifier into a simple sequence number
- Remove the Household identifier completely from the datarecord.

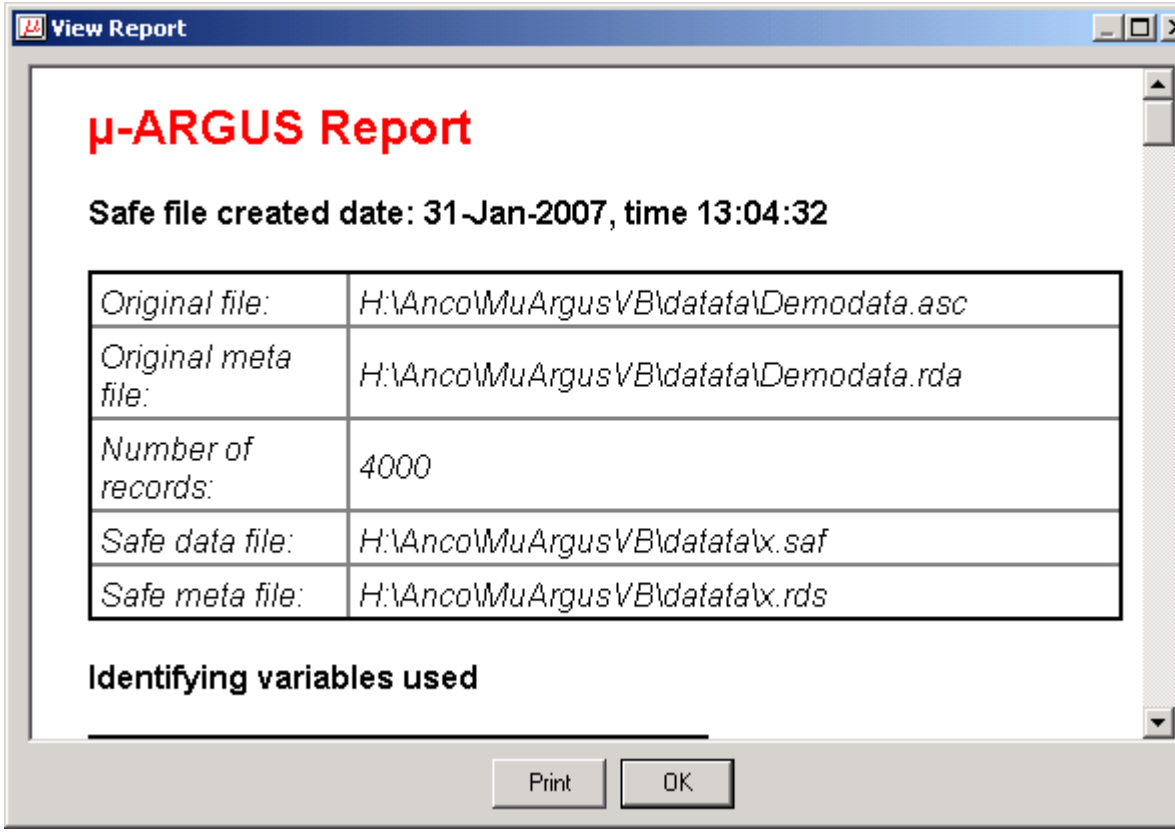
Finally there is an option to write the records in a random order. This could be done to prevent easy linking to other data files. Note that this option is available only for the version using a fixed format ASCII file

Pressing the 'make file' button will start the writing process. First you will have the opportunity to specify the name of the output file. Automatically a report file will be written at the same time. The name of the reportfile is the same as the outputfile, but with the extension .HTML. This HTML-format allows you to inspect the report later with a browser. When the datafile is ready, however, µ-ARGUS will show the report directly.

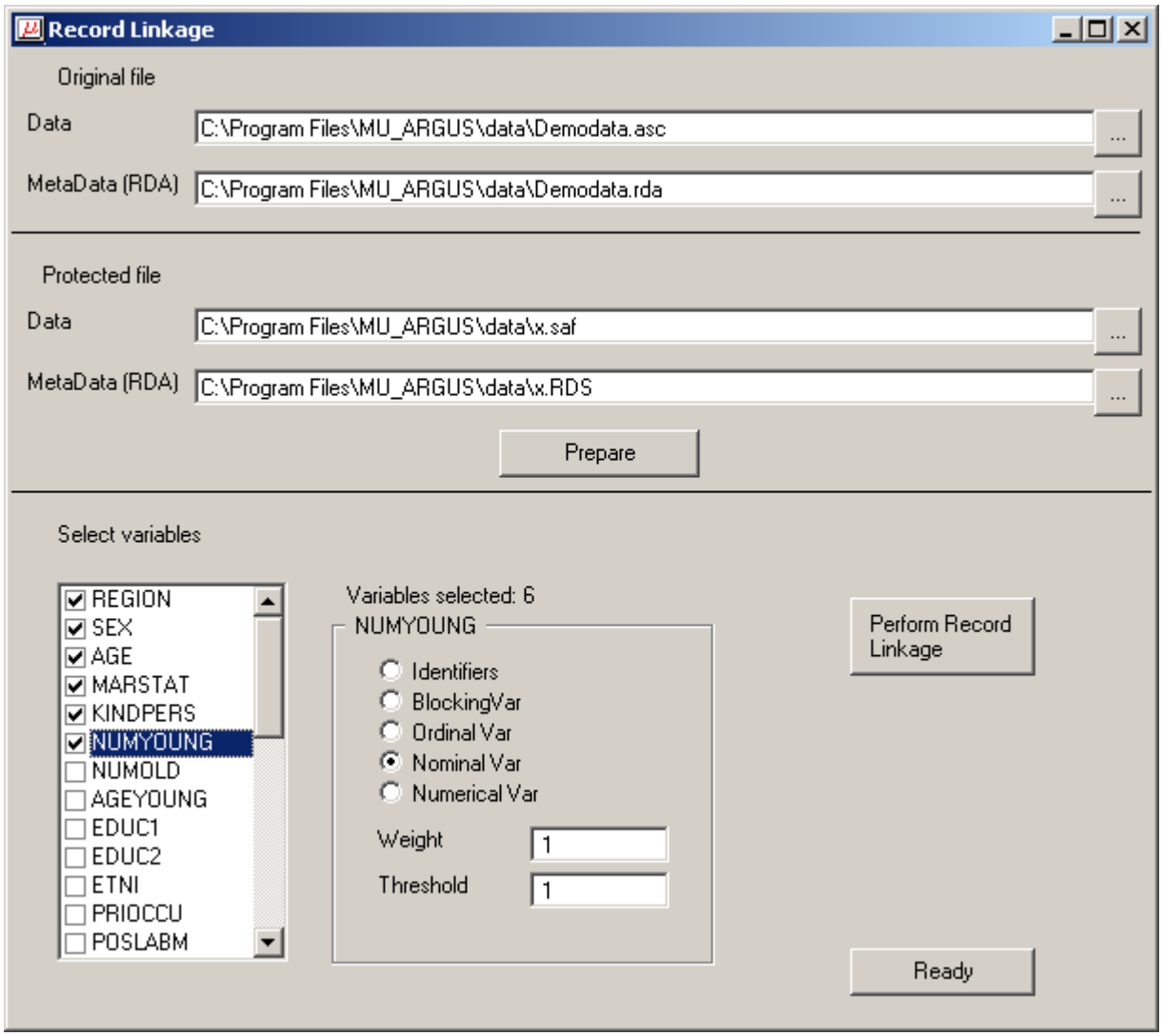
In any case you will now have a 'safe'file, safe according to the modifications that you have applied.

5.5.2 Output|View Report

Views the report file which has been generated with Output|Make suppressed file



5.6.1 Record Linkage



Record Linkage (see also section 3.2.9) can be used to assess the quality of certain Disclosure Control methods. A first version of Record linkage is included in μ-ARGUS.
First both files to be linked must be specified, including the record description on a familiar RDA-file. When both files have been specified you press the prepare button. If successful the second part of the screen will be filled and the variables can be selected. In addition the role of the selected variables can be specified.
The use of the blocking variable is recommended as the procedure will be very time-consuming for larger datasets. Not using a blocking variable may lead to excessive computing time.

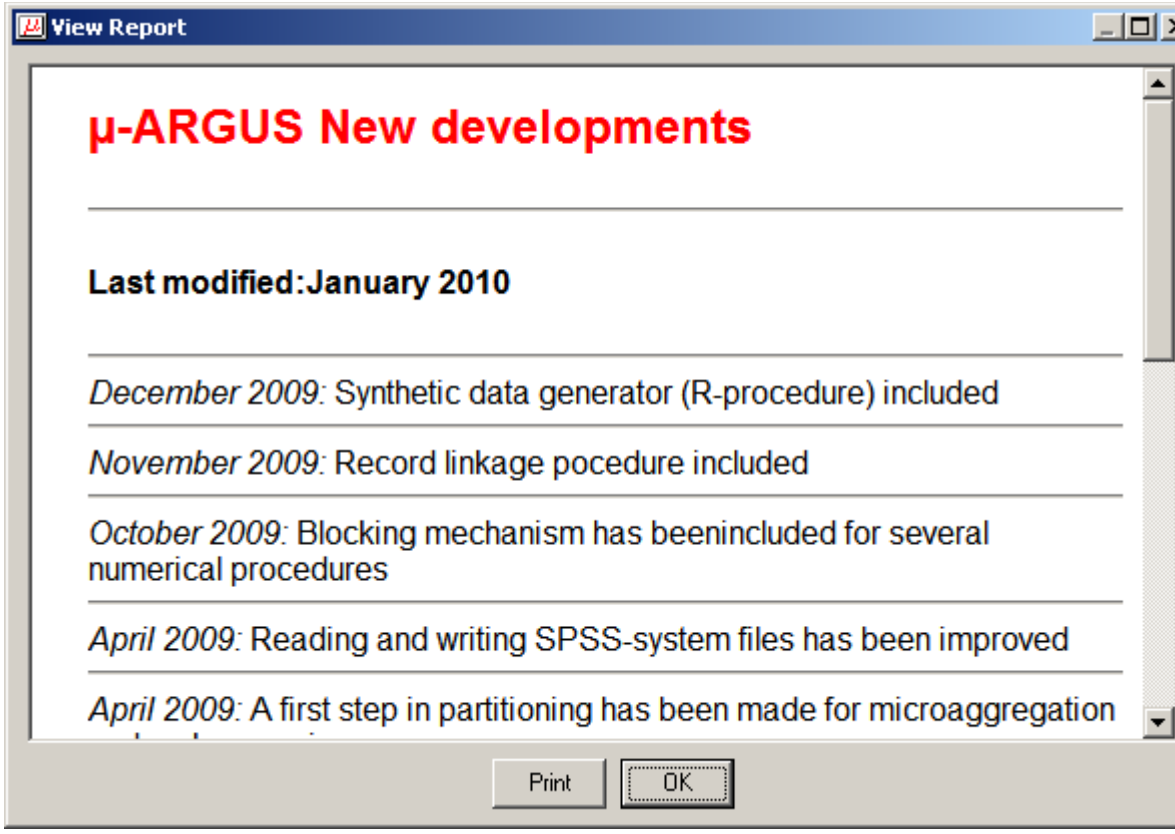
5.7 Help

5.7.1 Help|Contents

Shows the content page of the help file. This program has context-sensitive help. Whenever you press the F1, you will get some information referring to the stage of the program where you are.

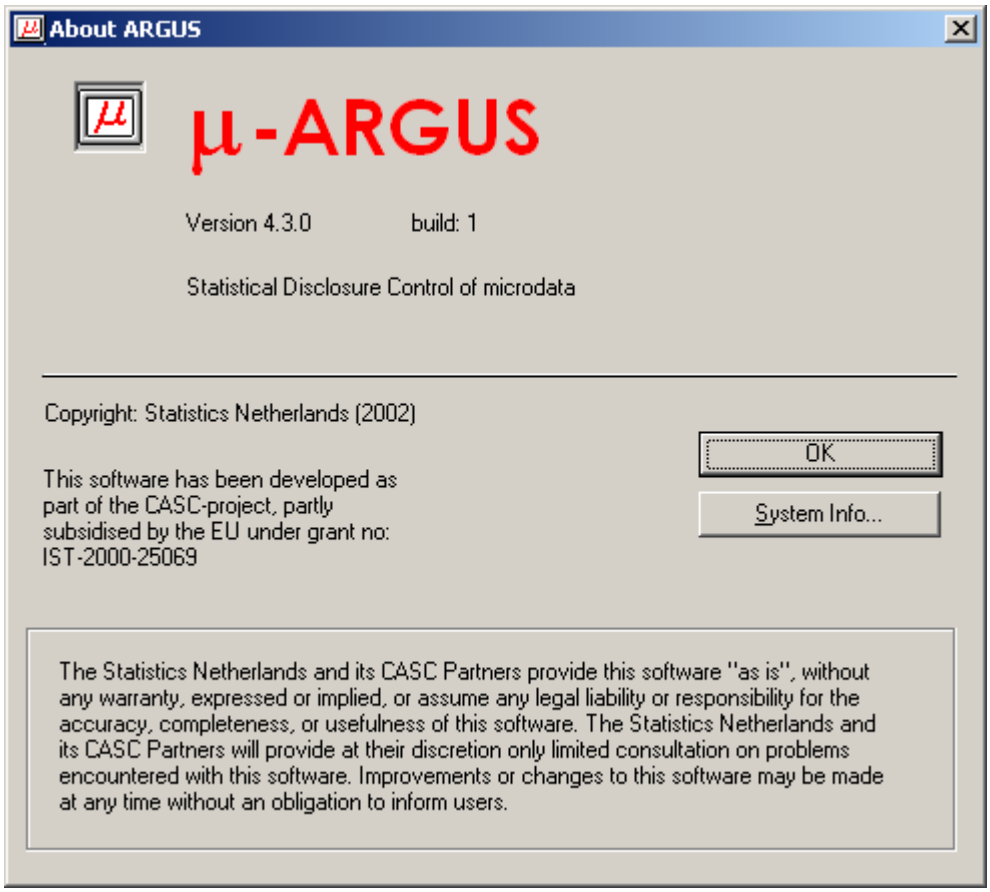
5.7.2 Help|News

An overview of the latest extensions and modifications of µ-ARGUS is given here.



5.7.3 Help|About

Shows the about box.


About ARGUS
μ-ARGUS
Version 4.3.0
build: 1
Statistical Disclosure Control of microdata
Copyright: Statistics Netherlands (2002)
This software has been developed as part of the CASC-project, partly subsidised by the EU under grant no: IST-2000-25069
OK
System Info...
The Statistics Netherlands and its CASC Partners provide this software "as is", without any warranty, expressed or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of this software. The Statistics Netherlands and its CASC Partners will provide at their discretion only limited consultation on problems encountered with this software. Improvements or changes to this software may be made at any time without an obligation to inform users.

6 FURTHER DECRIPTIONS

6.1 Meta data file

The metadata for μ-ARGUS is stored in a textfile. The default extension is .RDA. Via the interface you can enter and modify the metadata. See section 5.3.1.

We will describe the file here, if you want to create the .RDA file yourself.

The .RDA file starts with the specification of the different file types.

- Fixed format: No Specifications as it is considered the default
- Free format: The first line will be

```
<SEPARATOR> ","
```

- Free format with names on the first line: the first two lines will be

```
<SEPARATOR> ","
<NAMESINFRONT>
```

- SPSS: the first line will be

```
<SPSS>
```

After this introduction section all the variables will be described.

On the first line for each variable the name of the variable is specified, followed by the starting position in the datafile (fixed format only), the fieldwidth and optionally one or two missing codes. Note that for categorical variables at least one missing value is mandatory.

In the following lines additional information for the variable can be specified. The information in the format of a keyword (enclosed by '<', '>') followed by the relevant parameters.

• <RECODABLE>	This variable may be recoded.
• <CODELIST>	Name of the codelist file.
• <IDLEVEL>	Identification level. (0,1,2,3,…)
• <TRUNCABLE>	The codelist is hierarchical and therefore removing one or more digits is a relevant way of recoding.
• <NUMERIC>	The variable is numeric and rounding etc. is allowed.
• <DECIMALS>	The number of decimal positions for a (numeric) variable.
• <WEIGHT>	The variable contains the sample weights. Randomising (check) the weight is an option by specifying the amount of noise to add, as a percentage.
• <HOUSE_ID>	This variable is a household identification.
• <HOUSEHOLD>	A household variable contains typically the same value for each member of a household. When the suppression of the value for one member is necessary, it will be done for each member.

• <SUPPRESSWEIGHT>	Priority weight for the selection of the suppression pattern; default value = 50.
• <RELATED>	This specifies a relation between variables. If this variable will be suppressed, also the related to variable will be suppressed.

An example of a meta data file for fixed format:

```
REGION 1  4  9999  9998
  <RECODABLE>
  <CODELIST>  "Regio.cdl"
  <IDLEVEL>  1
  <SUPPRESSWEIGHT>  50
SEX 5  1 9
  <RECODABLE>
  <CODELIST>  "Sex.cdl"
  <IDLEVEL>  2
  <SUPPRESSWEIGHT>  50
AGE 6  2  99
  <RECODABLE>
  <IDLEVEL>  3
  <SUPPRESSWEIGHT>  50
  <NUMERIC>
……….
COMPCODE 31  3  999
  <RECODABLE>
  <IDLEVEL>  3
  <SUPPRESSWEIGHT>  50
  <TRUNCABLE>
……
WEIGHT 44  6
  <NUMERIC>
  <DECIMALS>  2
  <WEIGHT>
INCOME 50  8  99999999
  <NUMERIC>
```

6.2 Codelist file

Codelists can be specified for explanatory variables. The codes are stored in a separate file (default extension .CDL).

However the codes are only used to enhance certain windows during the processing. μ-ARGUS itself will create the coding schemes for the variables used during the processing of the datafile. So a code not specified in the .CDL-file will not cause any problem, only the label is not available. Also codes specified but not found in the data file will be ignored.

The structure of the file is simple. Each line contains a code and a label separated by a ","

region.cdl

```
  1,Groningen
  2,Friesland
  3,Drenthe
```

```
  4,Overijssel
  5,Flevoland
  6,Gelderland
  7,Utrecht
  8,Noord-Holland
  9,Zuid-Holland
 10,Zeeland
 11,Noord-Brabant
 12,Limburg
 Nr,North
 Os,East
 Ws,West
 Zd,South
```

6.3 Global recode file

Global recoding is a powerful method to reduce the number of primary unsafe cells. It reduces the size of the table, but the advantage is also that the number of primary unsafe cells is reduced. It is a classical balance to decide how far you should go when applying global recodes, but often the resulting table contains much more information, compared to a table with many, but suppressed cells.

For a hierarchical coding scheme τ-ARGUS allows recoding via collapsing the tree structure of the hierarchy. But for non-structured codelists the global recode must be specified manually

The structure is always: A new code is assigned to a set of old codes. So all the old codes are collapsed into the new code. A set can be either a list of individual codes, separated by a comma, or an interval indicated by a lower code, dash upper code. If the upper or lower code is not specified an open interval is assumed.

Examples:

For a variable with the categories 1,…,182 a possible recode is then:

```
1:  - 49
2: 50  - 99
3: 100 - 149
4: 150 -
```

This implies that every code below 49 will be recoded into the new code 1,all codes between 50 and 99 will be the new code 2 etc.

For a variable with the categories 01 till 10 a possible recode is:

```
1:  01 , 02
2:  03 , 04
3:  05 - 07
4:  08 , 09 , 10
```

An important point is not to forget the colon (:) if it is forgotten, the recode will not work.
Recoding 3: 05,06,07 can be shortened to 3: 05-07.

And the two different schemes can be combined as well

```
1: 02 – 06, 09
```

is a valid recode as well.

Additionally new values for the missing codes can be specified as well as a new codelist. See the example below.

Note that μ-ARGUS treats the codes as strictly alphanumeric. So '01' is really different from '1' and ' 1'.

Example of a .GRC-file

```
   20:1-15
   21:16-30
   22:31-45
   23:46-60
   24:61-75
   25:76-90
   26:91-105
   27:106-120
   28:121-135
   29:136-150
   30:151-165
   31:166-182

   <MISSING> 97 98
   <CODELIST>
regiop.cdl
```

7 INDEX

B
blocking.....84, 86
G
global recoding.....13, 29, 30, 34, 36, 46, 50, 52, 60, 61, 94
H
Hybrid data.....19
M
Microaggregation-based hybrid data.....20
Muralidhar-Sarathy hybrid generator.....19
P
partitioning.....84, 86
S
synthetic microdata.....18