CASC PROJECT

Computational Aspects of Statistical Confidentiality

30 January 2002

Strategy for the implementation of individual risk methodology into μ-ARGUS: case of independent and hierarchical units

Alessandra Capobianchi
Silvia Polettini
Maurizio Lucarelli

ISTAT, MPS/D
Via C. Balbo, 16
00184, Roma
Italy

Deliverable No: 1.2-D2

Preface

This deliverable contains all the information needed to implement in μ-Argus the individual risk for the hierarchical case. For reason of completeness the documentation has been merged with the material contained in the previous deliverable (Deliverable No: 1.2-D1).
In particular, we stress that the paper has been modified with major changes in Section 5 (5.2 added), 6 (6.3, 6.4 added) 7, 8, 9. Since the whole material has undergone a global revision we recommend reading the paper from the beginning including the Sections concerning independent record.

1. Introduction

In this report the individual risk estimation algorithm is presented, focusing on the main differences with respect to the current version of μ-Argus, and explaining what should be implemented. The next Section approaches the algorithm at a general, descriptive level. In Section 3 we describe the variables that are needed to implement the risk (key variables, special types variables, etc.). Section 4 explains how to evaluate the frequencies of combinations of key variables in the sample, f_k , and discusses the estimation of these frequencies in the population, $\hat{F}_k$. These two processes will be described also in the presence of missing values. In Section 5 estimation of the individual risk is presented. Section 6 contains the flow charts of the algorithms used for risk estimation and a few relevant remarks about

the risk. In Section 7 we show some graphs as examples of what could be useful. Section 8 describes how to tie the μ-Argus suppression strategy with our methodology, finally in Section 9 we describe how to produce a *safe file*.

2. Algorithm overview

Our approach is based on the need to handle *sample data*: the data file therefore does not include the whole population, but a subset of it, and every unit in the file represents one or more units of the population through the *individual weights*. So, the individuation and treatment of unique (or rare) combinations is no longer adequate in order to make the input file '*safe'*, but is necessary to deal with a method that considers the sampling aspect of the data set.

Our method estimates the level of disclosure risk for each unit, defined as the probability of identifying an individual. A schematic representation of the step to evaluate this risk is given in Figure 1.

After the application of the risk calculation algorithm, each record i will have associated its own value of the disclosure risk ρ_i. At this point, the user will input a threshold α, that he considers the maximum tolerable risk. This choice should be based on a graph representing the distribution of the individual risk in the file.

Once α has been selected, the algorithm will apply the suppressions only to records i such that $\rho_i > \alpha$, following a suppression method similar to the one already implemented in Argus.

At this stage the user should judge by analysing a report containing a summary of the suppressions introduced. By construction, for each individual in the output file, the disclosure risk will never exceed the threshold α. The user can further evaluate the overall post-suppression risk reduction by means of a graph. This means that the algorithm for calculating the individual risk has to run again on the output file.

He/she has now two choices: a) he is satisfied by the result, and the output file is recorded as *safe file*; b) he/she discards the results, choosing to rollback to the previous steps, e.g. selecting another level of α.

Finally, we must stress that our approach also deals with *hierarchical* files, i.e. when units are linked by a hierarchical relation, for example a household file where the units that belong to the same household share the same *household identifier*.

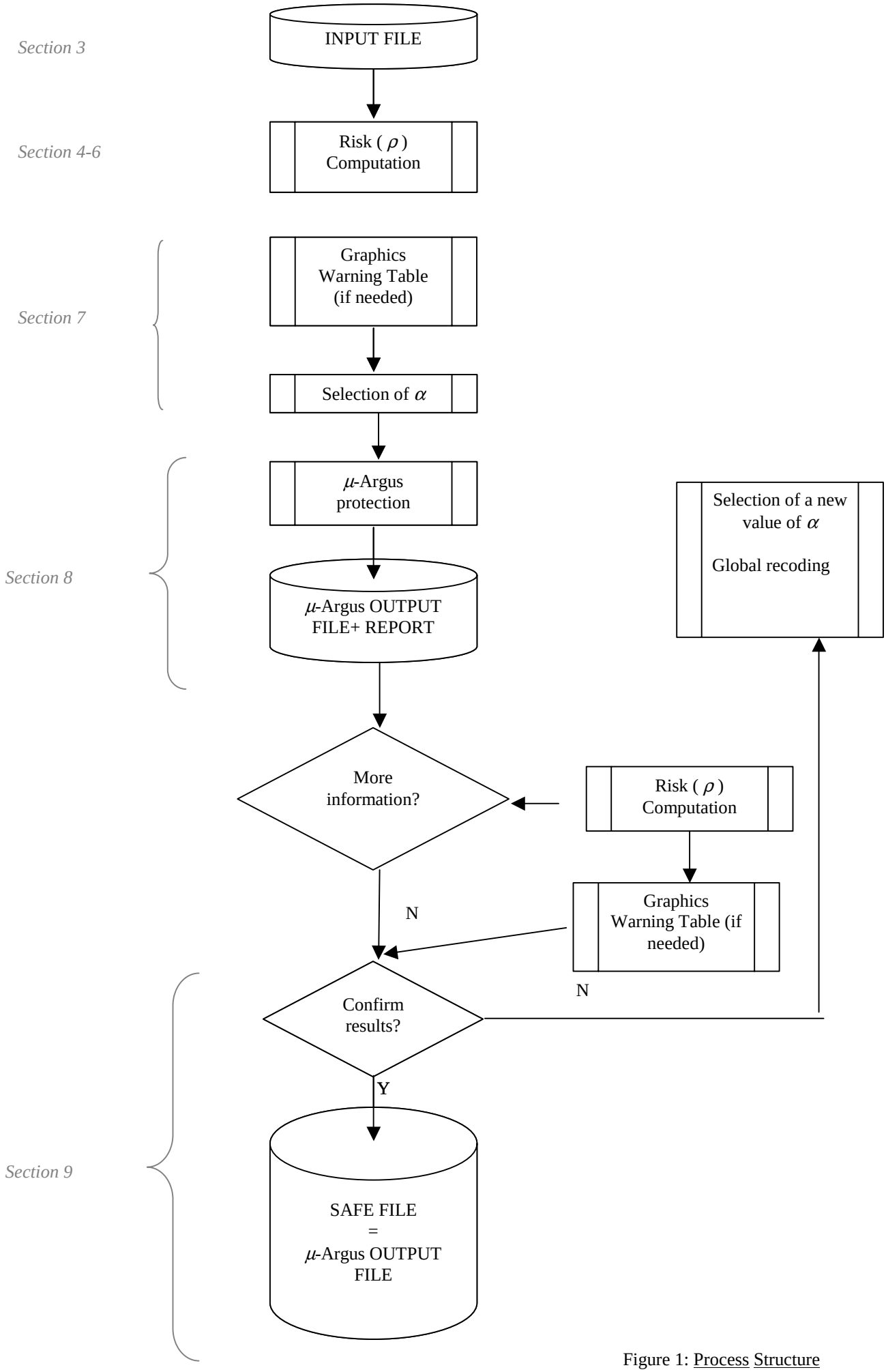


Figure 1: Process Structure

3. Input file

The file acquisition procedure is substantially the same as the one implemented in μ-Argus; it is however indispensable to get further information, additional to that already collected in the "specify metadat" μ-Argus window, in order to calculate the individual risk:

- Individual weights (w_i , always ≥ 1)
- Individual identifier (UnitID)

Moreover, in the hierarchical case:

- Household identifier (HHID)
- Household size (S): the number of units with the same HHID value; if this variable is not present in the input file, it is necessary to calculate it.

4. Frequencies calculation

A fundamental step for risk estimation is the computation of the frequencies f_k and $\hat{F}_k$.

First of all, we consider the population as partitioned into K sub-populations ($k = 1, \ldots, K$), defined through all the possible combinations of categories of the key variables.

It must be stressed that in the individuation of these sub-populations we use ***all*** the variables defined by the user as 'key'.

Suppose we have a file composed by 8 units:

HHID	UnitID	Key_Var1	Key_Var2	Key_Var3	Key_Var4	w_i	$f_i = f_{k(i)}$	$\hat{F}_k$
1	1	1	2	5	1	18	2	110
1	2	1	2	1	1	45,5	2	84,5
1	3	1	2	1	1	39	2	84,5
1	4	3	3	1	5	17	1	17
2	5	4	3	1	4	541	1	541
2	6	4	3	1	1	8	1	8
3	7	6	2	1	5	5	1	5
3	8	1	2	5	1	92	2	110

With $k(i) = k$ we denote the sub-population defined by the combination of categories of the key variables (*string*) in the unit i. In our example, there are 6 sub-populations, and unit 1 and 8 belong to the same sub-population identified by the string (1 , 2 , 5 ,1).

With f_k we represent the frequency (count) of units in the k^{th} sub-population that are present in the sample (i.e. in the file). The estimation of these frequencies in the population, $\hat{F}_k$, is given by the sum of the weights associated with the units belonging to that sub-population: $\hat{F}_k = \sum_{i:k(i)=k} w_i$.

In the example above, we get:

$$\text{k(1) = k(8) = (1, 2, 5, 1)} \Rightarrow \begin{cases} \hat{F}_{k(1)} = \hat{F}_{k(8)} = w_1 + w_8 = 18 + 92 = 110 \\ f_{k(1)} = f_{k(8)} = 1+1 = 2 \end{cases}$$

A problem may arise if there are missing values in the key variables.

Actually, a missing value could stand for any of the possible categories of the variable considered. Thus, in our opinion, computation of the f_k should take this into account. Consider the set of strings or combinations which are '*compatible*' with the one characterising the k^{th} sub-population, i.e. combinations which completely agree, except at most for one or more missing categories. In the presence of missing values, computation of f_k may be pursued by

counting the number of units having strings compatible with the k^{th} sub-population. A similar argument can be applied to $\hat{F}_k$.

The table below shows how missing values affect computation of the relevant quantities in the context of the previous example:

HHID	UnitID	Key_Var1	Key_Var2	Key_Var3	Key_Var4	w_i	$f_i = f_{k(i)}$	$\hat{F}_k$
1	1	1	2	5	1	18	3	149
1	2	1	2	1	1	45,5	2	84,5
1	3	1	2	.	1	39	4	194,5
1	4	.	.	1	5	17	3	576
2	5	4	3	1	.	541	3	566
2	6	.	3	1	1	8	2	549
3	7	6	2	1	5	5	2	22
3	8	1	2	5	1	92	3	149

The string (1 , 2 , . , 1), associated whit the UnitID 3, is compatible with the sub-populations identified by the strings (1 , 2 , 5 , 1) and (1 , 2 , 1 , 1), and, in the same way, in each of this two sub-populations it has to be counted also the unit characterised by the string (1 , 2 , . , 1).

So:

$\hat{F}_{k(1)} = \hat{F}_{k(8)} = w_1 + w_8 + w_3 = 18 + 92 + 39 = 149,$

$f_{k(1)} = f_{k(8)} = 1 + 1 + 1 = 3,$

while

$\hat{F}_{k(3)} = w_3 + w_1 + w_8 + w_2 = 39 + 18 + 92 + 45,5 = 194,5$

$f_{k(3)} = 1 + 1 + 1 + 1 = 3,$

5. Risk computation

The individual hierarchical risk to be associated with each unit, can be seen as the sum of two main factors:

$$r_i^{hier} = r_{k(i)}^{ind} + r_i^{dep} \tag{1}$$

In the following Sections we describe in more detail each risk component.

5.1. Base individual risk

The first component, $r_i^{ind} = r_{k(i)}^{ind}$, represents the base individual risk for a unit *i* having combination *k(i)=k* of key variables, and is the same for every unit belonging to the same sub-population. It is given by:

$$r_{k(i)}^{ind} = r_k^{ind} = \left(\frac{\hat{p}_k}{1-\hat{p}_k}\right)^{f_k} \left\{ A_0 \left(1 + \sum_{j=0}^{f_k-3} (-1)^{j+1} \prod_{l=0}^{j} B_l \right) + (-1)^{f_k} \log(\hat{p}_k) \right\} \tag{2}$$

where

$$\hat{p}_k = \frac{f_k}{\hat{F}_k} = \frac{f_k}{\sum_{i:k(i)=k} w_i}, \tag{3}$$

and w_i are the individual weights,

while $B_l = \frac{(f_k - 1 - l)^2}{(l+1)(f_k - 2 - l)} \frac{\hat{p}_k^{l+2-f_k} - 1}{\hat{p}_k^{l+1-f_k} - 1}$ and $A_0 = \frac{\hat{p}_k^{1-f_k} - 1}{(f_k - 1)}$. (4)

The above formulation works for $f_k \geq 3$; if $f_k = 1$ we use:

$$r_k = \frac{\hat{p}_k}{1 - \hat{p}_k} \log\left(\frac{1}{\hat{p}_k}\right), \tag{4a}$$

while if $f_k = 2$:

$$r_k = \left(\frac{\hat{p}_k}{1 - \hat{p}_k}\right) - \left[\left(\frac{\hat{p}_k}{1 - \hat{p}_k}\right)^2 \log\left(\frac{1}{\hat{p}_k}\right)\right]. \tag{4b}$$

However, we found the task of evaluating formula (2) exceedingly heavy or even absolutely impossible when observed frequencies are too large. In these cases the introduction of a numerical approximation is convenient. We obtained satisfactory results using:

$$r_k = \frac{\hat{p}_k}{f_k - (1 - \hat{p}_k)} \tag{5}$$

In the flow chart presented in Section 6 this approximation is used for frequencies greater than 40. We were forced to set this value because of software limitations: however, use of a higher threshold could increase precision. In the same flow chart are presented solutions for the two cases where the denominator is 0 in the two equations presented in formula (4) – i.e. $f_k = 1$ and $f_k = 2$.

5.2. Dependence risk

The last term of (1), r_i^{dep} , stands for the risk caused by the dependence structure shared by the units. The underlying idea is application of Boole's formula for mutually exclusive events. First of all, we now consider only the units inside an household (i.e. the records sharing the same HHID value, e.g. h^*).

The mathematical expression is:

$$r_i^{dep} = \left(1 - r_i^{ind}\right) \sum_{\substack{j:h(j)=h(i) \\ j \neq i}} \left(r_j^{ind} \prod_{\substack{l:h(l)=h(i) \\ l<j \\ l \neq i}} \left(1 - r_l^{ind}\right) \right) \tag{6}$$

where $\prod_{\substack{l:h(l)=h(i) \\ l<j \\ l \neq i}} \left(1 - r_l^{ind}\right) = 1$ when there is no value of l satisfying the condition $l < j$.

Next we show the procedure for evaluation of r_i^{dep} through some examples.

The calculation does **not** depend on the order in which the units sharing the same HHID appear in the file, therefore, to keep matter simple, we follow the order in which the units are recorded in the file. Let r_{1st}^{ind} (r_{2nd}^{ind}, r_{3rd}^{ind}, …) be the base individual risk of the first (second, third, ...) individual in the file having HHID = h^*. Let $S(i) = s$ be the size of the household containing the unit i.

If $S(i) = 2$, then the two individuals inside the household will have the following dependence risk:

if unit i is the first of the household, then: $r_i^{dep} = (1 - r_{1st}^{ind}) r_{2nd}^{ind} = (1 - r_i^{ind}) r_{2nd}^{ind}$;

if unit i is the second of the household, then: $r_i^{dep} = (1 - r_{2nd}^{ind}) r_{1st}^{ind}$;

If $S(i) = 3$, then the three individuals inside the household will have the following dependence risk:

if unit i is the first of the household, then: $r_i^{dep} = (1 - r_{1st}^{ind}) r_{2nd}^{ind} + (1 - r_{1st}^{ind})(1 - r_{2nd}^{ind}) r_{3rd}^{ind}$;

if unit i is the second of the household, then: $r_i^{dep} = (1 - r_{2nd}^{ind}) r_{1st}^{ind} + (1 - r_{2nd}^{ind})(1 - r_{1st}^{ind}) r_{3rd}^{ind}$;

if unit i is the third of the household, then: $r_i^{dep} = (1 - r_{3rd}^{ind}) r_{1st}^{ind} + (1 - r_{3rd}^{ind})(1 - r_{1st}^{ind}) r_{2nd}^{ind}$.

5.3. Final risk

Finally, in order to consider other factors influencing the risk (such as the quality of the key variables, the intruding probability, and so on) we use a multiplying factor π so the final risk formula is given by:

$$\rho_i = \pi * r_i^{hier} \tag{7}$$

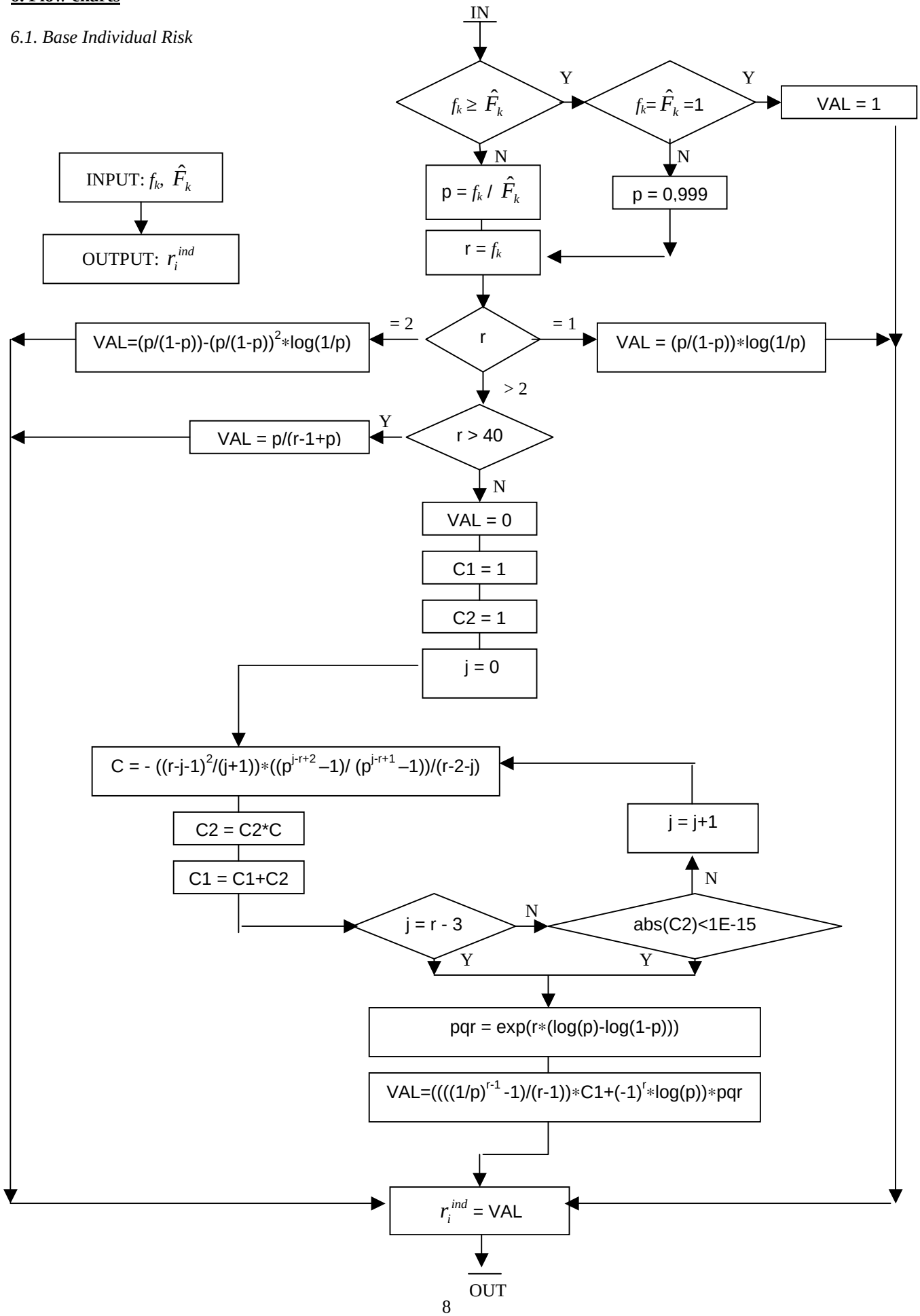
The factor π, set to 1 as the default, should be requested to the user by an interactive window before the risk computation starts.

We must say that if the input file is not hierarchical, the risk calculation is reduced to the application of (2), then adjusted by the π parameter:

$$\rho_i = \pi * r_{k(i)}^{ind} \tag{8}$$

6. Flow charts

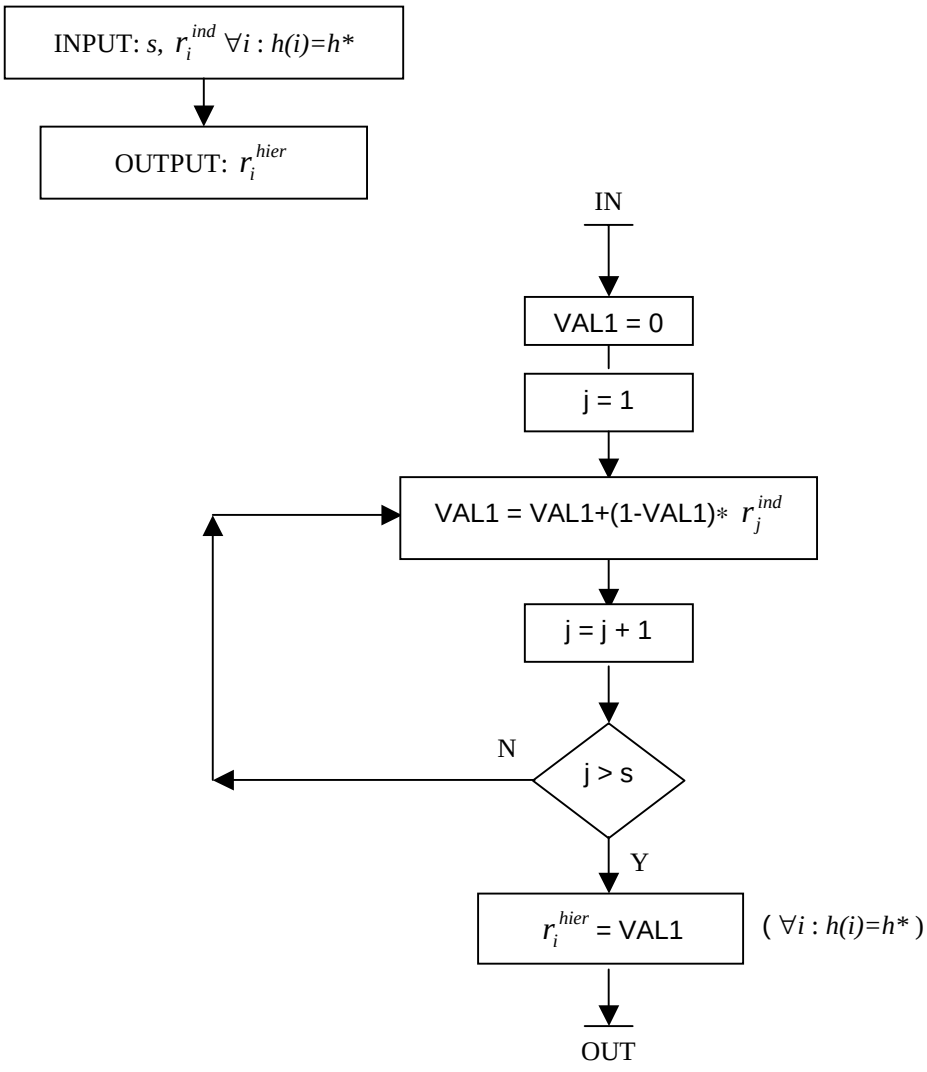
6.1. Base Individual Risk



The previous algorithm has to run for each record of the input file.

6.2. Hierarchical Risk

The following algorithm gives in output r_i^{hier} (i.e. the sum $r_i^{ind} + r_i^{dep}$), which is the same for each record sharing the same HHID value (i.e. $\forall i : h(i)=h^*$). Once a household has been selected, the index *j* goes from the first to the last (s^{th}) record of the household.



6.3. Final Risk

The final risk value (ρ_i) is obtained multiplying the output of the previous algorithm (r_i^{hier}) by the π parameter (see formulas (1), (7) in Section 5.) .

6.4. Remarks about the base individual risk

This Section will discuss some peculiarities of the base individual risk, which can be useful for the next developments.

As said before (Section 5.1.), r_i^{ind} is the same for each individual belonging to the sub-population $k=k(i)$ identified by the k^{th} combination of the categories of the key variables. This is the reason why we used the symbol $r_{k(i)}^{ind}$.
The definition of the base individual risk (see Section 4), makes use of the whole set of key variables.
Suppose we use only one key variable having *J* categories: the population will then be partitioned into *J* sub-populations; adding one or more key variables, the resulting partition will be a refinement of the previous one. In the latter case unit *i* will be more easily identifiable. Accordingly, the base individual risk will generally increase. In fact, r_i^{ind} is a nondecreasing function of the number of key variables.

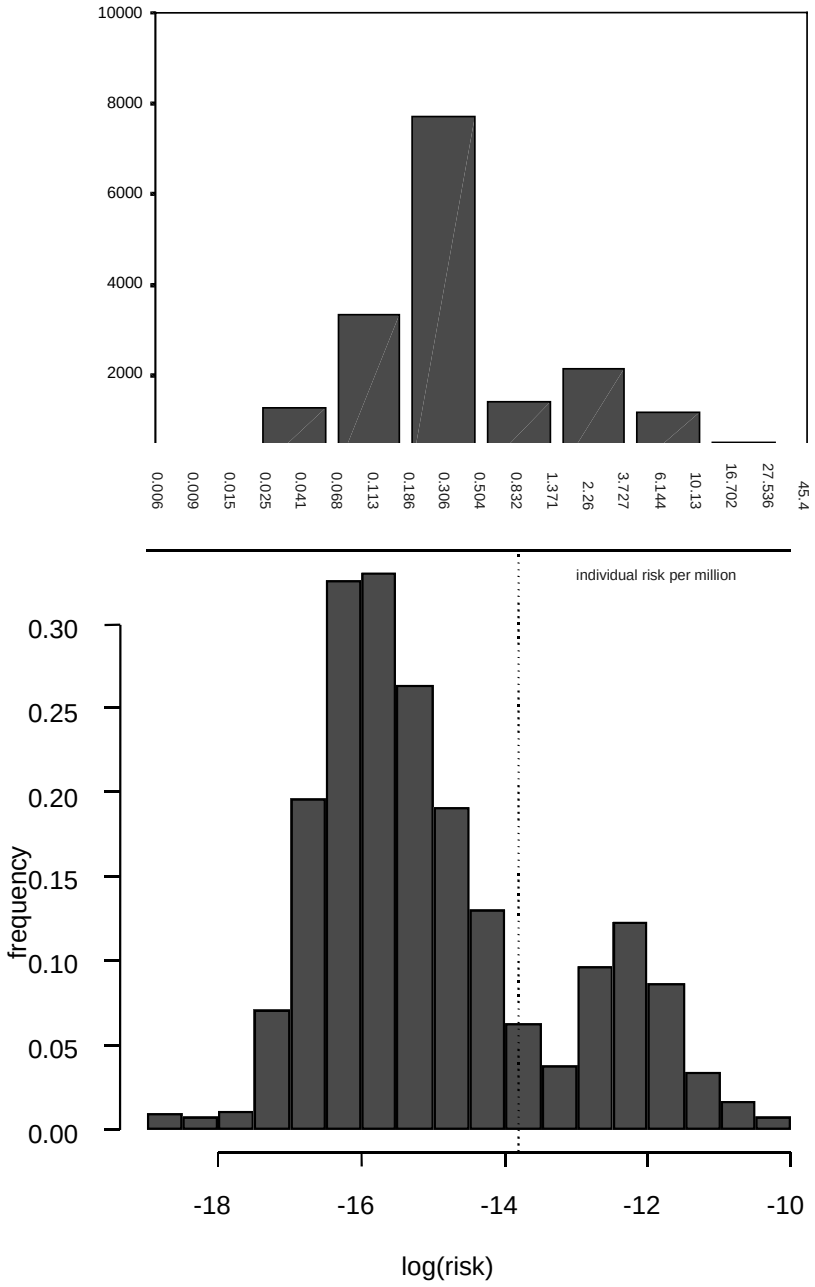
Let r_i^{ind} be the base individual risk calculate considering the whole set of key variables and $r_i^{ind}(s)$ (referred to as the *core risk*) the base individual risk computed using *S* (the household size) as the only key variable; then the considerations above imply that

$$r_i^{ind}(s) \leq r_i^{ind}.$$

Now, we know that if the file is hierarchical the variable *S* is always considered a key variable but it can never be suppressed since can be desumed as consequence of the hierarchical structure of the data. This means that the individual risk cannot be made smaller than the value of the core risk.
Hence, whenever $\pi * r_i^{ind}(s)$ exceeds the α threshold, we are dealing with a set of unsafe records, whose risk can by no means be lowered below the threshold by suppressing the other key variables. This is why a pre-screening of the core risk is suggested, as described in the following Sections, before Argus suppression procedure is started.

7. Graph

After the evaluation of the final risk ρ_i , the user needs a graphic to fix the threshold α. We think of it as a frequency histogram. By our experience, the graph could be clearer showing a logarithmic scale on the *x* axis (the one with the ρ_i values) or, which is the same, representing $log(\rho_i)$ instead of ρ_i . However, the labels on the axis should still report the corresponding ρ_i value, in order to better evaluate the appropriate α value.
Next we show, as examples, some risk graphs we used, though they do not perfectly correspond to the above description:



As remarked in Section 6.4., the core risk should be inspected before the file is protected.

For this purpose, a table containing the values $\pi * r_i^{ind}(s)$ for those household sizes *s* such that the threshold is exceeded, i.e. $\pi * r_i^{ind}(s) > \alpha$, could be a valuable tool. For such *s* values, the table should also report the counts and percentages of individuals and families in the data set.

For example, having fixed a threshold $\alpha = 0.5*10^{-6}$, suppose that $\pi * r_i^{ind}(s) > \alpha$ for 75 individuals out of 57.000 (belonging to seven families out of 20.000). Then the user could be warned against this situation with a message of the following form:

> *Warning: Identification risk for one or more individuals cannot be reduced below the fixed level* <(show current alpha>.

followed by a report showing the details below:

$\alpha = 0.5*10^{-6}$					
S	$\pi * r_i^{ind}(s)$ *(risk per million)*	*Counts (individuals)*	*% (individuals)*	*Counts (families)*	*% (families)*
13	0.8581	13	0.0228	1	0.005
11	0.6922	22	0.0386	2	0.010
10	0.5441	40	0.0701	4	0.020
total		75	0.1315	7	0.035

It would be useful to have both the graph and the warning message in the same window in which the user chooses the α value, so that as the value of α changes, the vertical line of the threshold shifts on the histograms and the warning message, if any, is refreshed.

8. Application of μ-Argus

After the final risk (ρ_i) has been evaluated for each record and the value of α has been chosen, the protection step follows through the local suppression method.

As far as we know, in μ-Argus an optimised procedure is implemented, based on minimisation of the suppressions in the unsafe combinations. A combination is considered unsafe if it occurs not more than D_k times in the data set, where D_k is the *threshold* value.

First, the procedure generates the combinations to be inspected following two possible alternatives: a) using the identification levels, b) generating all tables up to a given dimension. Then, after the unsafe combinations have been found, the procedure checks the presence of unsafe combinations in each record and chooses the suppression which minimises the number of suppressions (see 'μ-Argus ver. 2.5 User's Manual'; de Waal – Willenborg: 'Minimizing the Number of Local Suppression in a Microdata Set' - Proj M1-79-589, First Draft, May 31, 1994).

For the implementation of our methodology, we need to introduce some adjustments in μ-Argus protection strategy.

First of all, the identification rule must be changed: a *combination* of key variables is considered *unsafe* if the final risk ρ_i of an individual having that combination of attributes exceeds a given threshold α, which means that the D_k criterion used in μ-Argus is no more adequate.

Second, unsafe combinations are progressively identified via generation of all tables of any dimensions, which must proceed from dimension one up to the highest (*K*, the number of key variables in the data set)[1].

Notice that if a string is found unsafe, any string which contains the latter will be unsafe as well (see Section 6.4.).

For the actual selection of the individuals at risk, we need to distinguish between independent and hierarchical files, as described in the following two Sections.

After the unsafe strings are singled out, the same protection algorithm already implemented in μ-Argus can be applied, producing the *μ-Argus output file.*

[1] To reach this aim with μ-Argus ver. 2.5 we used either the identification levels (specifying for each key variable a different identification level) or the generation of all tables up to a given dimension (the highest).

8.1. Independent file

Recall that in this case the final risk is $\rho_i = \pi * r_{k(i)}^{ind}$.

As reported in Section 6.4., the base individual risk r_i^{ind} and hence ρ_i is nondecreasing in the number of key variables used for identification.

This allows us to apply the checking procedure starting from the *K* univariate contingency tables (step 1). The final risk is evaluated at each category of each of the *K* key variables. If the current value of ρ_i (based on one key variable only) exceeds α for a category, this category is considered unsafe and moreover each combination of key variables containing such category will be unsafe as well. Having selected only the current (step 1) safe strings, the algorithm proceeds in screening pairs of categories of key variables (step 2), identifying the unsafe pairs and so on, adding one dimension a time, up to the highest (step *K*). At each step *k*, the combinations containing a substring judged unsafe at step *k-1* are not screened, as they are certainly unsafe.

Alternatively, instead of the final risk ρ_i, the screening algorithm may check the individual risk r_i^{ind}, and compare it with the threshold α/π.

8.2. Hierarchical file

Recall that for hierarchical files we refer to $\rho_i = \pi * r_i^{hier} = \pi * \left(r_{k(i)}^{ind} + r_i^{dep}\right)$.

Notice that in the hierarchical case *S* cannot be suppressed; hence the core risk $r_i^{ind}(s)$ is a lower bound for r_i^{ind}, as $r_i^{ind}(s) \le r_i^{ind}$ (see Section 6.4.). This implies that the first variable to be inspected at step 1 by the algorithm is household size, *S*; moreover, this priority helps us in the construction of the warning table described in Section 7.

Of course, if the file is not hierarchical, then there is no problem in suppressing values of *S* (which plays no special role); therefore, the warning table should never be built in case of independent files. Adoption of the above mentioned priority for variable *S* has no influence in the selection of unsafe strings for independent records. Hence the latter can represent a generalised screening strategy.

For the sake of simplicity, let us consider the hierarchical risk of an individual *i* belonging to a household of size 3, say, having HHID(*i*) = h^*. As described in Section 5, each household member shares the same hierarchical risk $r_{h^*}^{hier} = r_{3rd}^{hier} = r_{2nd}^{hier} = r_{1st}^{hier}$, defined in terms of the base risk of each individual as follows:

$$r_{h^*}^{hier} = r_{3rd}^{hier} = r_{3rd}^{ind} + (1 - r_{3rd}^{ind}) r_{1st}^{ind} + (1 - r_{3rd}^{ind})(1 - r_{1st}^{ind}) r_{2nd}^{ind} =$$

$$= r_{1st}^{ind} + r_{2nd}^{ind} + r_{3rd}^{ind} - r_{1st}^{ind} r_{2nd}^{ind} - r_{1st}^{ind} r_{3rd}^{ind} - r_{2nd}^{ind} r_{3rd}^{ind} + r_{1st}^{ind} r_{2nd}^{ind} r_{3rd}^{ind}$$

(the right-hand side of the equation shows that referring to the third individual in the group causes no loss of generality, as the risk is in fact the same for each of the three members of the household).

Being by definition $0 \le r_i^{ind} \le 1$, the previous equation shows that $r_{h^*}^{hier} \le r_{1st}^{ind} + r_{2nd}^{ind} + r_{3rd}^{ind}$. Analogously, denoting by $\rho_{h^*}^{hier}$ the final hierarchical risk of any individual belonging to household h^*, we have $\rho_{h^*} = \pi * r_{h^*}^{hier} \le \pi * \left(r_{1st}^{ind} + r_{2nd}^{ind} + r_{3rd}^{ind}\right)$.

Hence if each of r_{1st}^{ind}, r_{2nd}^{ind} and $r_{3rd}^{ind} < \frac{1}{\pi}\frac{\alpha}{s}$, then certainly we have $r_{h^*}^{hier} < \frac{\alpha}{\pi}$, e.g. $\rho_{h^*} < \alpha$.

Consequently, we suggest screening the independent risk r_i^{ind} by the same algorithm explained in Section 8.2. for independent records, with the only differences that the threshold changes to $\frac{1}{\pi}\frac{\alpha}{s}$ and ***S* cannot be suppressed**.

Note that this procedure may lead to an overprotected file, but in our opinion this proposal could be a good compromise, as it involves the concept of safe/unsafe combinations which is already used in μ-Argus. A possible solution to the problem of overprotection could be the following:

at the last step *K*, for an individual *i* whose base risk r_i^{ind} exceeds the threshold, evaluate the hierarchical risk of his/her household $h(i)=h^*$ before the protection algorithm starts, and apply the suppressions in the usual way only if $r_{h^*}^{hier} < \frac{\alpha}{\pi}$.

For those records whose *core* risk exceeds the given threshold (cfr. Section 6.4.), no standard suppression can in principle be applied using the algorithm described above; consequently, for those records we suggest suppressing the whole set of key variables (except *S*, which would be useless). Yet the records above will remain unsafe at a second screening: the user will decide by him/herself if this strategy is satisfactory; otherwise, the only available option is to increase the threshold or exclude those records from the output file.

9. Safe file

Once the suppression algorithm has been applied the user can judge the results by inspection of a window reporting a summary of the suppression procedure. In the hierarchical case, since the output file is overprotected, the risk calculation algorithm (Section 5) should run again on the output file produced by μ-Argus, in order to produce the new values of the risk after the protection step. Next, the graph representing the current risk distribution (Section 7) have to be shown.

At this point the user can check the protection level attained, and he has two options:

a) *confirm*: the output file is recorded as the *safe file*;

b) *rollback*: the user is not satisfied by the results. He/she is now presented with different options, which can be applied one by one or in combination. He can: specify a different α value, or otherwise recode some variable, or both.