18

**CSO Best Practice for Statistical Disclosure Control of Tabular Data**

**Timothy Linehan and Karina Dineen**

# Contents

[0](#_Toc527645835)

[Contents 1](#_Toc527645836)

[Executive Summary 3](#_Toc527645837)

[1. Introduction to Tabular SDC 4](#_Toc527645838)

[1.1 Background 4](#_Toc527645839)

[1.2 What is disclosure? 4](#_Toc527645840)

[1.3 Illustration of disclosure, suppression and information loss. 5](#_Toc527645841)

[2. Safety of Tables and Suppression 7](#_Toc527645842)

[2.1 Safe and unsafe output 7](#_Toc527645843)

[2.2 Example of safe/unsafe output and primary disclosure. 8](#_Toc527645844)

[2.3 Illustration of Secondary Disclosure and Suppression 11](#_Toc527645845)

[2.4 When to apply tabular-SDC? 12](#_Toc527645846)

[2.4.1 Output checking of RMFs 12](#_Toc527645847)

[2.4.2 Checking CSO Statistical Output 13](#_Toc527645848)

[2.5 Linked tables 13](#_Toc527645849)

[3. Tabular-SDC Primary and Secondary Suppression Rules 14](#_Toc527645850)

[3.1 Primary suppression rules 14](#_Toc527645851)

[3.1.1 Minimum Frequency Rule 14](#_Toc527645852)

[3.1.2 (n , k) Dominance Rule 15](#_Toc527645853)

[3.1.3 p% Rule 16](#_Toc527645854)

[3.2 Secondary Suppression methods 17](#_Toc527645855)

[3.2.1 Data manipulation/programming based methods. 17](#_Toc527645856)

[3.2.2 Hypercube 18](#_Toc527645857)

[3.2.3 Optimal 18](#_Toc527645858)

[3.2.4 Modular 18](#_Toc527645859)

[3.2.5 “SIMPLEHEURISTIC” 19](#_Toc527645860)

[4. Tabular-SDC software in the CSO 20](#_Toc527645861)

[4.1 Tau-Argus 20](#_Toc527645862)

[4.2 R (sdcTable) 20](#_Toc527645863)

[4.3 SAS-based implementations 21](#_Toc527645864)

[4.4 Visual Basic method for Microsoft Excel. 22](#_Toc527645865)

[5. Tabular-SDC examples 23](#_Toc527645866)

[5.1 Illustration of primary and secondary suppression using Tau-Argus 23](#_Toc527645867)

[5.1.1 Tau-Argus operation 23](#_Toc527645868)

[5.1.2 Microdata file preparation 24](#_Toc527645869)

[5.1.3 Metadata file preparation 24](#_Toc527645870)

[5.1.4 Reading the file into Tau-Argus 25](#_Toc527645871)

[5.1.5 Preparing tables for Primary Suppression. 25](#_Toc527645872)

[5.1.6 Applying secondary suppression 28](#_Toc527645873)

[5.2 Demonstration of primary and secondary suppression using R 32](#_Toc527645874)

[5.2.1 R File Structures 32](#_Toc527645875)

[5.2.2 Data is read into R 32](#_Toc527645876)

[5.2.3 Primary suppression in R 33](#_Toc527645877)

[5.2.3 Secondary suppression in R 34](#_Toc527645878)

[5.2.4 Production of output file 34](#_Toc527645879)

[5.3 Using the Visual Basic approach 35](#_Toc527645880)

[5.3.1 The script 35](#_Toc527645881)

[5.3.2 Operation of Script 35](#_Toc527645882)

[6. Recommendations 36](#_Toc527645883)

[6.1 Recommendations 36](#_Toc527645884)

[7. Further reading 36](#_Toc527645885)

# Executive Summary[[1]](#footnote-1)

The document provides guidance on the theory, implementation and CSO policy implications of Tabular Statistical Disclosure Control (Tabular-SDC). This relates mainly to tabular output files created by the CSO and by researchers from Researcher Microdata Files (RMFs), and how they may be processed to avoid the unintentional disclosure of sensitive information to researchers.

In Chapter 1, the main concepts in Tabular-SDC are discussed. The concepts of disclosure and information loss are illustrated.

In Chapter 2, the concepts of safe and unsafe output are introduced. Primary and secondary suppression are illustrated using hypothetical examples.

In Chapter 3, the theory behind primary and secondary suppression is explained. The methods used are outlined.

Chapter 4 features a detailed discussion of the various software options available to the CSO for implementing tabular-SDC.

Chapter 5 shows detailed practical examples of the use of Tabular SDC software for synthetic CSO datasets. Both Tau-Argus and R are shown to carry out the necessary steps. Detailed guidance is given on the practical implementation of both software packages. It should be noted that the use of Tau-Argus and R does not mean that these are the only way of implementing Tabular SDC.

Chapter 6 contains recommendations made for Tabular SDC within the CSO as well as suggested policy points. These are:

1. The final decision on what constitutes acceptable levels of disclosure risk remains with data custodians. The rationale for SDC decisions within business areas must be recorded and retained. The SDC risk from unauthorised matching should also be considered.
2. There are numerous software solutions for implementing Tabular SDC, including Excel, SAS, Tau-Argus and R, and custodians may choose which to use. The criteria of a successful solution is whether it, in the informed opinion of the data custodians, reduces disclosure risk to acceptable levels.
3. While final decisions on Tabular SDC belong to data custodians, the SDC process can be assigned to designated staff within a section or even division. The role of the custodian may be supervisory in these cases.
4. A successful SDC implementation in an area may in some, but possibly not all cases, reduce the burden on individual sections, since the risky manual output checking process will hopefully be streamlined or replaced by an appropriate software implementation. None of this is guaranteed however.

# 1. Introduction to Tabular SDC

**Note: This document is one of two providing guidance on Statistical Disclosure Control (SDC) within the Central Statistics Office (CSO). It provides guidance on applying SDC to CSO tabular output and incorporates international best practice.**

## 1.1 Background

Statistical Disclosure Control (SDC) refers to methods that allow the dissemination of statistical information while ensuring that individuals are protected against disclosure. The key challenge in SDC is achieving this protection while ensuring that information loss is kept to a minimum. SDC methods can be applied either to microdata files being provided to researchers (Microdata-SDC) or to tabular output. This document covers Tabular-SDC.

In recent years the CSO has faced both increasing demand for statistics from users and stakeholders and increasing obligations to ensure the confidentiality of statistical data. In particular, with the advent of the European Union’s GDPR (General Data Protection Regulation), there is a new requirement for the CSO to ensure that “appropriate safeguards” are in place for statistical data. This is in addition to the existing legal obligations of the CSO, under the Statistics Act, to preserve the confidentiality of respondents[[2]](#footnote-2). These safeguards include both methods to ensure the protection and confidentiality of data during the storage and processing stage as well as the importance of insuring that published data is non-disclosive.

By tabular output, we refer to tables produced as the output of the statistical analysis of tables. For the CSO, these can either be produced by the CSO itself or by researchers from CSO-issued microdata. Of particular concern is frequency-based tabular data(counts) or magnitude-based (totals). It should be noted that more complex types of tabular output, such as predictive output generated from models, is usually considered to be of lower statistical risk.

There is a danger with tabular output, both from administrative and survey sources, that if tables are sufficiently detailed, then there may be a significant risk of individuals being identified.

## 1.2 What is disclosure?

Disclosure is when data issued by the CSO, either as an RMF or as a table, allows users to learn previously unknown information about respondents, be they individuals or organisations, in statistical surveys or administrative datasets. There are two main types of disclosure risk:

* *Identity disclosure -* if a respondent can be identified with a disseminated data record or table entry containing confidential information. For example, if a table showed that there was 1 death in a particular town, then an intruder could identify the deceased from the table.
* *Attribute disclosure* – this refers to the case where an intruder could determine the attributes of an individual based on information in the released data. As a hypothetical example, if a table disclosed that all individuals in a particular location commuted for four hours a day, then an intruder would know that attribute about all residents in that location.

Disclosure events are classified into primary or secondary disclosure.

* *Primary disclosure –* When an unsafe cell is published.
* *Secondary disclosure –* if an unsafe cell is removed, but its value can be deduced from the aggregate totals and the other non-suppressed values.

Disclosure can also be described as either exact or approximate.

* *Exact disclosure* – when the exact value of an attribute can be determined by a user.
* *Approximate disclosure* – when a user can determine the approximate value of a respondent value. This may be more difficult to prevent than exact disclosure. A detailed example of approximate disclosure is shown in Section 3.1.2.

The sensitivity of a variable means whether it is sufficient to guard against exact or approximate disclosure of the variable value.

## 1.3 Illustration of disclosure, suppression and information loss.

Consider the following case. The CSO publishes a table with a county-level breakdown of high-net-worth individuals by nationality. The following facts apply.

* High net-worth Individuals, in this scenario, are defined as having a net worth of greater than 100 million euros.
* There are only two nationalities – Irish and Ruritanian.
* There are 2 Ruritanian citizens living in Ireland.
* There are 500 high-net-worth individuals in Ireland, with the following nationality breakdown.

**Table 1 – No. of high net-worth individuals by nationality.**

|  |  |  |
| --- | --- | --- |
| **High Net-Worth Individuals** | | |
| **Irish** | **Ruritanian** | **Total** |
| 499 | 1 | 500 |

This is an example of **primary disclosure** of an attribute. The Ruritanian high net-worth individual, upon seeing the table, will be aware that he has been identified – his income has been disclosed to the public.

Next, consider the case where the CSO suppresses the unsafe cell. The act of removing the unsafe cell is called **primary suppression.**

**Table 2 – No. of high net-worth individuals by nationality. Primary suppression.**

|  |  |  |
| --- | --- | --- |
| **High Net-Worth Individuals** | | |
| **Irish** | **Ruritanian** | **Total** |
| 499 | X | 500 |

This table is not any safer for the following reasons: The aggregate of 500 individuals is known (the total value) ; the value for the number of Irish individuals is known (499) and the the missing value for Ruritanians is simply 500-1 = 499.

This is an example of **secondary disclosure**.

Next, a decision is made to also supress the Irish figure. This is **secondary Suppression**:

**Table 3 – High-net worth individuals by nationality. Primary and secondary suppression.**

|  |  |  |
| --- | --- | --- |
| **High Net-Worth Individuals** | | |
| **Irish** | **Ruritanian** | **Total** |
| X | X | 500 |

As a result, the Ruritanian value can no longer be calculated. The tabular output in Table 3 can be considered to be safe. While somewhat extreme, this illustrates how care needs to be applied in the production of tabular output.

Next, the question of information loss information loss:

* Table 1 does not have any information loss. On the other hand, it is highly disclosive.
* Table 3 has removed the disclosure risk. On the other hand, the (useful and safe) information on the number of Irish respondents has not been published. This is an example of Information loss.

Therefore, it is necessary to balance both disclosure protection and information loss. Numerous SDC rules and methods have been developed to address these issues.

# 2. Safety of Tables and Suppression

## 2.1 Safe and unsafe output

Usually, the CSO publishes two types of tabular output:

* Frequency tables
  + The frequency or count of a categorical variable
  + Example- The number of individuals with third-level qualifications in a particular area.
* Magnitude tables
  + The aggregate of a numerical variable
  + Example - The Gross Value Added (GVA) to the economy.

All tables are cross-tabulations of categorical variables and can be published/disseminated at various levels. For example, GVA could be disseminated in the following table combinations:

* GVA at a national level
* GVA by NACE Type
  + GVA by NACE Division
  + GVA by NACE Rev 2 code).
* GVA by NUTS region
  + GVA by NUTS-2
  + GVA by NUTS-3
* GVA by NUTS and NACE
  + GVA by NACE and NUTS-2
  + ……
  + GVA by NACE Rev 2 and NUTS-3

Some of these tables are more safe than others. For example, publishing GVA at a national level (a single figure) contains no risk of identifying any information about individual respondents. A safe table. or entry therein, is one where there is a negligible risk of disclosure.

On the other hand publishing GVA by NACE Rev 2 and NUTS-3 could potentially include numerous unsafe cells – i.e. cells from which information on respondents could be confirmed. This is because there may be numerous cells where there are very few numbers of respondents. Unsafe cells are those where there is a significant risk of identifying respondents. or deducing attributes about respondents.

As a general rule, the more levels and categories in the cross-tabulations that compose a particular table, the greater the risk of unsafe cells existing. Once unsafe cells have been identified, they can be made safe using SDC methods.

## 2.2 Example of safe/unsafe output and primary disclosure.

This scenario demonstrates the subtleties of safe/unsafe output and primary disclosure.

* Ireland has 100 companies involved in mining and quarrying.
  + This is NACE code B (Mining and Quarrying).
  + Two types of mining - 0510 Mining of hard coal & 0721 Mining of Uranium
  + Uranium mining is responsible for over 90% of NACE code B sales.
* Ireland has 2 companies involved in Uranium mining.
  + These companies have NACE code B
  + The NACE Rev 2 code is 0721 – Mining of Uranium.
  + This is a secretive industry (neither company wants it known that it is involved in this business – this is a sensitive attribute).
* Zamyatin Industries is Ireland’s main miner of Uranium.
  + It gets 95% of Sales Revenue from Uranium in Ireland
  + It is unaware that it has any competitors (the mining is a secretive business)
  + It is based in the NUTS 3 Region of Midland (IE063).
  + It is based in the NUTS 2 Region of East and Midlands (IE06).
* Kerdashev Technology is Ireland’s other miner of Uranium
  + It produces 5% of Irish Uranium (and gets 5% of sales).
  + It is unaware that it has a competitor.
  + It is based in the NUTS 3 region of Mid-West (IE051)
  + It is based in the NUTS 2 Region of IE05 (Southern)

Consider Table 4. This is a safe frequency table with no risk of disclosure.

**Table 4 Example of a safe frequency table.**

|  |  |
| --- | --- |
|  | **B - Mining and Quarrying** |
| **NUTS-2 Region** | **Number of enterprises** |
| IE05 - Southern | 50 |
| IE06 - East and Midlands | 50 |
| Total | 100 |

The next step is to consider is to produce a more detailed table (by NACE Rev 2). Table 5 now unintentionally discloses that two Uranium mining firms exist and also reveals their location. In this case, there are two primary disclosures, each with a value of 1.

**Table 5 Illustration of unsafe frequency table**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **B - Mining and Quarrying** | |  |
| **NUTS-2 Region** | **0510 Mining of hard coal** | 0721 - Mining of Uranium | Total |
| IE05 - Southern | 49 | 1 | 50 |
| IE06 - East and Midlands | 49 | 1 | 50 |
| Total | 98 | 2 | 100 |

It is then decided that the detailed geographical breakdown is leading to excessive disclosure risk. Therefore a new table, featuring a detailed breakdown of NACE group B is produced, but only at a national level.

**Table 6 Illustration of national aggregation**

|  |  |  |
| --- | --- | --- |
|  | **B - Mining and Quarrying** | |
| **NUTS-2 Region** | **0510 Mining of hard coal** | 0721 - Mining of Uranium |
| Total | 98 | 2 |

From a frequency perspective, this still leads to a situation where each firm becomes aware of the other’s existence. Is there still a high level of primary disclosure risk? Yes. In such cases, it is best to have a larger acceptable minimum frequency – a minimum of 5 respondents is a value frequently cited in business statistics. The following section contains a detailed discussion of Minimum Frequency Rules.

The next case to consider are magnitude figures - Sales aggregates. As noted, Zamyatin Industries receives 95% of Irish Uranium Sales. Table 7 demonstrates an unsafe magnitude table.

**Table 7 Example of an unsafe national total.**

|  |  |
| --- | --- |
|  | **B - Mining and Quarrying** |
| **NUTS-2 Region** | **0721 - Mining of Uranium Revenue (€)** |
| Total | 100,000,000 |

Superficially, this table is safe. Very few users, outside of the enterprises, would know the ratio or volumes of sales in each enterprise. **However, the perspective of the respondents themselves must always be considered.** From this table, since both respondents are aware of their own sales revenues, each firm can work out how much revenue is represented by remaining firms. This situation is exacerbated if Table 5 has also been published - since each firm also knows that there is only one other firm. The combination of Tables 5 and 6 will allow each firm to calculate the revenue values of its rivals.

The next scenario to be considered is a national total for NACE Group B (Table 8). This table is also potentially disclosive, since the scale of Uranium mining is publicly known:

**Table 8 Example of potentially disclosive national total.**

|  |  |
| --- | --- |
|  | **B - Mining and Quarrying** |
| **NUTS-2 Region** | **Total Revenue (€)** |
| Total | 105,000,000 |

Since, it is publicly known that the Uranium industry dominates NACE Group B, it would be possible for well-informed users of statistics to make a reasonable guess as the total sales of Uranium in Ireland. Each firm, in addition, could approximately infer their revenue and that of their main competitor. **Therefore, particular care must be taken with magnitude figures where dominance is likely to be an issue.**

In conclusion, tabular output must be examined in detail to establish if it is safe or not. Safety rules are applied to identify primary disclosure risk are discussed in Chapter 3.

## 2.3 Illustration of Secondary Disclosure and Suppression

Consider Table 9 (a version of Table 4 where primary suppression has been applied). Though primary suppression has been applied, secondary disclosure still occurs:

**Table 9 Version of Table 4 where primary suppression has been applied**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **B - Mining and Quarrying** | |  |
| **NUTS-2 Region** | **0510 Mining of hard coal** | **0721 - Mining of Uranium** | **Total** |
| IE05 - Southern | 49 | a | 50 |
| IE06 - East and Midlands | 49 | b | 50 |
| Total | 98 | c | 100 |

* From the national total, (c), the overall number of 0721 – Mining of Uranium firms can be deduced as 2 (c = 100-98 = 2)
* For each NUTS-2 region, the number of Uranium firms can be deduced as 1 in each region (b = 50-49 = 1 and c = 50-49 = 1).

So, if no secondary suppression has been applied, it is extremely easy to deduce the missing values – this is secondary disclosure. To prevent this, secondary suppression can be applied. However, it is important to note that a badly-applied secondary suppression pattern can be of little-to-no value in preventing secondary disclosure.

Now consider the case where the CSO decide to apply secondary suppression by suppressing all figures for NUTS-2 Region IE06 – East and Midlands, in addition to the existing primary suppressions (Table 10). These are marked d and e.

**Table 10 Demonstration of bad secondary suppression.**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **B - Mining and Quarrying** | |  |
| **NUTS-2 Region** | **0510 Mining of hard coal** | **0721 - Mining of Uranium** | **Total** |
| IE05 - Southern | 49 | a | 50 |
| IE06 - East and Midlands | e | b | d |
| Total | 98 | c | 100 |

All the missing values can be deduced as follows:

* The missing total for IE06 (d) can be deduced: 100-50 = 50.
* The missing total for 0510 (e) can be deduced: 98-49 = 49.
* The remaining values (a,b,c) can then be calculated via 50-49 = 1 etc..

This scheme did not work because it failed to take account of the fact that aggregate and marginal totals (the National and NUTS/NACE respectively) were present. A safer example of secondary suppression in this case would simply be to recode the data to remove the NUTS and NACE subcategories (i.e. Table 4). As a result, there would be no risk of secondary disclosure.

## 2.4 When to apply tabular-SDC?

Tabular-SDC is to be applied in two situations. Firstly, when checking output generated by researchers from CSO-issued RMFs to ensure there is no disclosure. This is a process called “Output checking”. Secondly, when examining CSO tabular output intended for dissemination to ensure that there is no statistical disclosure.

### 2.4.1 Output checking of RMFs

Generally, CSO data custodians need to examine tabular output produced by researchers from RMF before approving the release/removal of this output from CSO control. When performing these tasks, CSO staff need to be aware of the following factors:

* Output checking needs to be applied to both frequency and magnitude tabular output.
* If the tabular output is a result of a non-linear process (such as predictions obtained from a model; model output or residuals; or other diagnostic data from a modelling process) then it may not be necessary to apply output checking to the data. The reason for this is that such processes can’t be easily inverted (to determine the values of the sensitive input data).
* Both primary and secondary suppression will need to be applied. Primary suppression on its own will probably not be sufficient.
* Custodians will have different thresholds for acceptable minimum frequency. For example, certain business areas may specify a very high minimum frequency, possibly on confidentiality grounds, or also on the grounds of a quality – in the sense that researchers may draw untenable conclusions based on very small cell sizes. Other areas may simply be concerned with ensuring a minimum sample size.
* It is suggested that areas with a significant burden in output checking of researcher tables consider Microdata-SDC methods such as recoding or perturbative (noise-adding) methods. The addition of noise will mean that some categories will have different values to their actual value. While this is not currently used by the CSO in its own statistical output, perturbation is a perfectly acceptable method for preparing researcher microdata. By using these methods when preparing the microdata file, much of the subsequent output checking work can be avoided (at a cost of increased information loss). For more information, refer to the CSO Guidance on Microdata Statistical Disclosure Control.
* The CSO has a Visual Basic script that allows rapid primary and secondary suppression when checking researcher output (See Sections 3.2.1, 4.4 and 5.3 for details of its operation. It has been designed for the bulk checking of output generated by researchers. This method can be applied to any researcher frequency table and greatly reduces the SDC burden in output checking. However, due to the fact that it leads to potentially heavy information losses, based on the secondary suppression method used, it is not recommended for tabular-SDC of CSO tables being disseminated.

### 2.4.2 Checking CSO Statistical Output

Generally, this will be applied to statistical output produced by the CSO for the following purposes:

* Tables produced by the CSO for dissemination to the public in statistical releases
* Tabular data available on statbank
* Tabular data produced in response to queries from stakeholders
* Data requested by Eurostat and other international agencies.

Often, the need for tabular-SDC can be limited in the first three cases, through the appropriate design of tables (in particular the use of recoding to remove sensitive categories). This may be simpler than applying tabular-SDC. However, tabular-SDC is often required for Eurostat data, since there is no often no flexibility in the data being sought. Both primary and secondary suppression are required if tabular-SDC is to be applied to CSO tabular output. If users have SDC-related questions, they are advised to contact Methodology.

## 2.5 Linked tables

It is common for tables to be linked. For example, there may be three tables with “debit”, “credit” and “balance”, where there is the following relationship: Debit+ Credit = Balance.

For such tables, tabular-SDC requires particular caution. For example, if primary (and secondary) suppression is applied to the “Debit” table, but not to the corresponding rows in the “Balance” and “Credit” table, then the suppressed values may still be determined by a user, thus leading to disclosure.

Applying SDC to linked tables usually involves the use of Tau-Argus or the R sdcTable package, though it is also possible to implement a solution to such a problem in SAS on a case-by-case basis.

# 3. Tabular-SDC Primary and Secondary Suppression Rules

In this section, there is a detailed discussion of the most common types of safety (otherwise known as sensitivity) rules – these are methods for determining if tabular output is safe or not.

## 3.1 Primary suppression rules

### 3.1.1 Minimum Frequency Rule

This is the most common primary suppression rule used in the CSO. It is applied to frequency tables. It has the following steps:

* Frequency tabular output is generated.
* A particular cell is deemed unsafe if the number of elements/contributors to the cell is less than a pre-specified value.

Table 11 illustrates this. A table is produced cross-tabulating employee type by hours worked. The minimum suppression rule has a pre-specified value of 5. If a cell contains less than 5 employees, it is to be suppressed.

**Table 11 Illustration of primary suppression for table based on numbers of employees**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Pre-Suppression** |  |  |  |  |  |
| **Employee Type** | **Over 40 hours** | **20-40 hours** | **10-20 hours** | **<10 hours** | **Total** |
| Supervisory personnel | 18 | 15 | 18 | 12 | 63 |
| Line personnel | 1 | 17 | 11 | 3 | 32 |
| **Total** | 19 | 32 | 29 | 15 | 95 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| **Post Suppression** |  |  |  |  |  |
| **Employee Type** | **Over 40 hours** | **20-40 hours** | **10-20 hours** | **<10 hours** | **Total** |
| Supervisory personnel | 18 | 15 | 18 | 12 | 63 |
| Line personnel | X | 17 | 11 | X | 32 |
| **Total** | 19 | 32 | 29 | 15 | 95 |

It is very important to note that the minimum frequency to be chosen is at the discretion of data custodians. While a sample cell size of 30 may be commonly cited in social statistics, that is due to the separate issue of the requirement that inferences from survey cells be based on sufficiently large numbers, thus ensuring high quality estimates. There may even be cases (in particular at national level) where values of 1 may be acceptable (such as crime and vital statistics). However, all rationales must be clearly recorded.

This example also re-iterates that primary suppression on its own is often insufficient. The missing values in this example can be inferred from the tabular totals.

### 3.1.2 (n , k) Dominance Rule

This is applied to magnitude tables. It is a good example of a method that protects against approximate disclosure. An example of approximate disclosure is shown in Table 12.

**Table 12 Illustration of approximate disclosure**

|  |  |  |
| --- | --- | --- |
| **Respondent:** | **Revenue (€)** | **Disseminate Status** |
| *Company A* | 50,000 | Confidential |
| *Company B* | 41,000 | Confidential |
| *Company C* | 1,000 | Confidential |
| *Company D* | 500 | . |
| *.* | . | . |
| *.* | . | . |
| *Company Z* | 500 | Confidential |
| **Cell Total** | 100,000 | Published |

Firstly, it is not possible to work out the precise values of each correspondent from the published cell total. However, we can see that Company A and B between them are responsible for 91% of the total (€91,000 out of €100,000) in the published cell. Especially when the dominant positions of Company A and B are publicly known, then there is an approximate disclosure of the Revenue values provided by A and B. For example, if Company A looks at the revenue total, it knows its own revenue of 50,000. It likely knows that Company B is of similar size to it and can thus obtain an approximate upper limit for the revenue of Company B.

The dominance rule addresses this. The (n,k) dominance rule means that primary suppression should be applied when the sum of the n largest contributors exceeds k% of the cell total.

The (n,k) dominance rule has the following formula:

Where is the cell aggregate, is the ith contributors value, n and k are the values defined in the safety rule..

A common implementation is called the (2,80) – in this case primary suppression is to be applied (the cell is unsafe) if the 2 largest contributions are greater than 80%. In the case of Table 12, two companies (A and B) contribute 91% of the magnitude in this cell. Therefore, under this rule the cell would be suppressed, thus preventing approximate disclosure. This approach can be implemented relatively simply. However, it still does not eliminate the need for secondary suppression (since otherwise the suppressed cell total can be easily re-obtained).

It should also be noted that the (n,k) rule is mathematically related to the p% dominance rule via the formula p = (100-k)/k

### 3.1.3 p% Rule

This is also used to address approximate disclosure. The p% rule means that a cell is sensitive (i.e. disclosive) if the cell total minus the two largest contributor is less than p% of the largest contributor. The p% rule can be formulated as follows:

The operation of the rule is shown in Table 13, which follows on from the example in Table 12. In this case we use a p% value of 25%, which should produce equivalent results to the (2,80) magnitude test using the relationship from the previous section, p = (100-k)/k= (100-80)/80 = 25%

**Table 13 Illustration of the p% rule.**

|  |  |
| --- | --- |
| **Total** | €100,000 |
| *Company A Revenue* | €50,000 |
| *Company B Revenue* | €41,000 |
| *Total - (A)-(B)* | €9,000 |
| *p% = 25% of Company A Revenue* | €12,500 |

So 9,000 (Total – (A)-(B)) is less than €12,500, indicating that the cell is disclosive and should be suppressed.

What is the logic behind this approach? It considers the perspective of the most well-informed intruder – this is the second largest respondent to the cell. Using the cell total (publicly available) and their own information, then they can make relatively close estimates as to the values of other respondents. If these values are unacceptably close (based on the p% - i.e. p% largest contributor), then the cell is suppressed.

This rule can be implemented using SDC software such as R or Tau-Argus.

## 3.2 Secondary Suppression methods

Of course, primary suppression is only the beginning. If only one primary cell is suppressed and if marginal or overall totals are included in the table, then the suppressed cell can be easily determined. Secondary suppression (see Section 2.3) is also required.

The main problem with secondary suppression is that it is a much more complex problem than primary suppression. Therefore, it involves complex programming, or numerical methods (in particular linear programming methods), in order to obtain a solution that protects the supressed primary cells while minimising information loss.

### 3.2.1 Data manipulation/programming based methods.

Secondary suppression can be performed in SAS or other database languages. If a cell is marked as suppressed and its relationships (either within/or without hierarchies) can be specified by the programmer, then it is possible for programming code to be developed in order to identify and suppress potentially unsafe secondary relationships.

This approach has a couple of advantages:

* While currently implemented in Visual Basic, this approach may be ported to other languages such as R.
* In Excel, it is possible to use a simple, albeit crude, Visual-basic based method based on applying primary suppression to disclosive values and applying secondary suppression to their nearest neighbours. This method can be applied to any table that can be read in Excel. The details of how this method are implemented is shown in Section 4.4 and 5.3. This method allows for rapid and safe checking of researcher output. However, due to the heavy levels of information loss it can produce, it is not recommended for tabular-SDC of official CSO tabular output. It should be possible to port this methodology to other languages if required.
* The code is usually prepared by someone who has extensive experience in the dataset being processed. They can easily identify and account for the relationships in the data that could lead to secondary disclosure. In effect, a bespoke solution that carefully considers the nature of the SDC problem for the data is prepared.
* Programming based approaches can deal with a serious shortcoming in SDC software – the inability to deal with **nested hierarchical tables**. For example, dedicated SDC software, such as Tau-Argus, cannot be applied to a table where the same value for Ireland occurs multiple times, but at different levels of a hierarchy (if Ireland is at the second level of a level-1 sub-total called “Europe” but also simultaneously at the fourth level of a level-1 sub-total called “World”). The only way to deal with such tables is via programming based solutions.

This approach has the following disadvantages however:

* These solutions will cover “exact” disclosure. However, they don’t address issues such as “approximate disclosure” (see 3.1.2).
* While the Visual Basic based method can be applied to any table, and is thus highly flexible, the nearest-neighbour based approach can lead to high levels of information loss.
* In SAS, deriving an appropriate solution, from first principles, can be very time consuming.
* Better solutions that balance SDC with information loss are often obtained via dedicated SDC software.

### 3.2.2 Hypercube

This is implemented in open-source (in Tau-Argus) using the GHMITER implementation. This approach is fast and, in general, produces very safe results, and covers both approximate and exact disclosure. It can be applied to hierarchical and non-hierarchical structures. The algorithm works as follows:

* The n-dimensional tables of interest are broken into a set of n-dimensional, unstructured tables.
* Each subtable is protected iteratively.
* For each suppression in the sub-table, all possible hypercubes that have this cell as a corner point are generated.
* After this process, a hypercube with minimum information loss is selected.

This method has the following characteristics. Computationally it is very fast, though not always suitable for very large tables. It is also prone to over-suppression, especially when compared to other methods. Its use is particularly suitable when reducing secondary disclosure risk is prioritised above minimising information loss.

### 3.2.3 Optimal

This method can be implemented in R or in Tau-Argus using the commercial FICO solver - see Section 4. This is a method which involves a classic linear programming operation (i.e. the SDC problem is considered as a set of linear inequalities that can be solved). It works as follows:

* If a cell has been primarily suppressed (i.e. marked as safe in the primary suppression stage), it is marked as 1.
* If the cell is safe, it is marked as 0.
* The optimal method uses the “branch and cut” operations research algorithm to produce suppression patterns.
* The one with the least information loss is then obtained.

The optimal method has the following characteristics:

* Greatly reduced information loss, compared to Hypercube, while producing safe output.
* This method is much more computationally intensive than Hypercube.
* Due to the complexity of the method, it cannot be applied, due to computing constraints, to very large series.
* It protects against both approximate and exact disclosure.
* The commercial solver required for its use in Tau-Argus is very expensive.

### 3.2.4 Modular

This method is available in R (sdcTable), Tau-Argus (via the open-source or commercial FICO solver) and SAS/OR. This technique breaks down the hierarchical table into several non-hierarchical tables. Each table is protected using a linear programming solver. protects them using LP-solver and then the smaller tables are combined to produce the protected table. It protects against both exact and approximate disclosure. However, the commercial solver is, as noted before, very expensive.

### 3.2.5 “SIMPLEHEURISTIC”

This is a method developed by Statistics Austria and implemented in the R sdcTable package. This method allows the very rapid (even for large datasets) application of secondary suppression to prevent exact disclosure. The main advantage of this approach is that it is a free and open source method that can be applied to very large datasets with extremely fast execution times. The main disadvantage is that it protects against exact disclosure only (though protecting against exact disclosure is often only what is required).

# 4. Tabular-SDC software in the CSO

## 4.1 Tau-Argus

Tau-Argus is an open-source software package, hosted by Statistics Netherlands. Arguably, it could be considered the standard package for Tabular Statistical Disclosure Control. It has the following features:

* Open source implementation of several methods including Hypercube and Modular method.
* It has a relatively easy-to-use graphical interface.
* Commercial solver required for the Optimal method (or for more computationally intensive modular calculations).
* Can be applied to both unaggregated microdata and to tables.
* Auditing function, to determine if unsafe cells still present after secondary suppression.
* Ease-of-use in implementing both primary and secondary suppression.
* Detailed manuals and documentation are available.
* It has provisions for the management of linked tables.

Tau-Argus has, however, numerous issues from an implementation perspective:

* The commercial solver is extremely expensive since it is only sold on a per-machine basis
* Error information is frequently cryptic.
* The manual preparation of metadata files is required, furthermore, such files must be prepared very carefully as otherwise errors will be triggered.
* It requires a non-CSO-standard implementation of Java, which means that machines upon which it is installed cannot run common Corporate management tools.
* Most seriously, the program cannot take account of complex tabular structures (such as nested hierarchies)- as a result it can mistakenly classify such tables as safe when they are not the case.

Tau-Argus is best used in the following cases: Flat tabular structures or small tables with simple hierarchical structures

In section 5.1, the use of Tau-Argus on a synthetic CSO dataset is demonstrated.

## 4.2 R (sdcTable)

The R package sdcTable contains a fully open-source implementation of Tabular-SDC. It has the following features:

* Includes a high-speed solver (SIMPLEHEURISTIC) for implementing secondary suppression of large datasets without the need for any commercial solver.
* Includes fully open-source and free implementations of Hypercube, Optimal and Modular methods. However, the use of these features is no longer recommended.
* It has been used and deployed by various NSIs successfully.
* There are no non-standard IT requirements – all that is required is a relatively current version of R.
* It has provision for linked tables
* It incorporates auditing.
* There is extensive documentation, including code examples.

It has the following disadvantages however:

* Successfully usage requires an, albeit limited, knowledge of R – it is not as easy to use as Tau-Argus
* It is also unable to manage unusual table structures (nested hierarchies etc).
* Documentation recommends that results be checked to ensure that resulting output is safe.

R sdcTABLE is recommended in the following cases: (a) Flat tabular structures; (b) small tables with simple hierarchical structures or (c) where the user wishes to use the optimal solver method (since this is the only open-source implementation available).

A demonstration of the use of sdcTable is shown in Section 5.2.

## 4.3 SAS-based implementations

In many areas of the CSO, tabular-SDC is applied via SAS (See Section 3.2). This includes various methods:

* Implementation of primary suppression rules in SAS code – this by itself is not sufficient.
* Implementations of primary and secondary suppression in SAS. These approaches are developed and tested by business areas and have the advantage of being tailored to the tables in question.
* Bespoke solutions, such as the one developed by Methodology for nested hierarchies, or multi-level hierarchies throughout the office.

As noted in Section 3.2, SAS based approaches have the benefit of data flexibility – the resulting solutions can handle complex data structures or complex relationships which would make use of R or Tau-Argus based approaches to be disadvantageous.

The main disadvantages have already been discussed – the lack of portability, lack of validation methods and the difficulties in balancing disclosure control and information loss. However, SAS based solutions, where already implemented, frequently work well.

It should be noted that if areas are already successfully using SAS-based methods for Tabular-SDC, and they are happy with the results in terms of balancing information loss with disclosure risk, then there is no need to migrate to R or Tau-Argus.

## 4.4 Visual Basic method for Microsoft Excel.

This is the method referred to in Sections 2.4.1 and 3.2.1. It is a simple Visual Basic script and works as follows. It is used to identify disclosive frequency counts.

* It is a Visual Basic script that can be run across all active worksheets in the table of interest
* The researcher specifies the minimum frequency. This is the threshold value and is determined by the researcher.
* There are no requirements as to the structure of the tables.
* In effect all the script does is look for cells less than the threshold value specified.
* These cells are suppressed and marked with “P”.
* The script then searchers for neighbouring cells (can be specified – usually at least one cell above or below and at least one cell to the left or the right. However, the script can be easily amended to suppress all neighbouring cells.
* These cells are then suppressed and marked with the letter “S”.
* The level of suppressing of nearest neighbours can be be adjusted by the programmer. However, it is necessary that at least two of the nearest neighbours are suppressed.

This approach has the following advantages:

* It is highly flexible
* It is easy to use
* It makes no requirements for pre-processing of the data of interest.
* A more sophisticated script would require much greater standardisation of output tables being submitted for checking.
* It is possible to translate this approach to other languages, such as R, if required.

It also has some disadvantages:

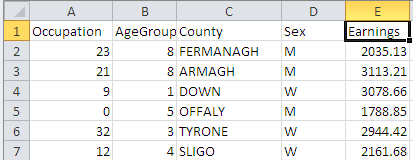
* Applies to frequency data only.
* The script does not distinguish between integers and rational numbers (hence it will apply itself to all the numbers in the table and suppress them.
* Since the nearest neighbours are suppressed based simply on their location in the table, and not based on their value, it means that useful data will be suppressed
* It can only protect against exact information loss, since it takes no cognisance of the nature of the table.

# 5. Tabular-SDC examples

## 5.1 Illustration of primary and secondary suppression using Tau-Argus

In this example, a CSO synthetic dataset (synth2) is examined via Tau-Argus. Primary and secondary suppression is carried out. The dataset has 60,500 records (Fig. 1 shows the structure), is comma-delimited and contains the following variables of interest:

1. Occupation
2. Sex
3. Age
4. County
5. Earnings



**Fig. 1 Synthetic dataset for tabular-SDC**

Here, we wish to examine if tables produced from this dataset may require primary and secondary suppression, and, if so, apply these operations via Tau-Argus.

### 5.1.1 Tau-Argus operation

While Tau-Argus can also examine individual tables, it is much more efficient to examine the microdata file (and possible tables) for potentially unsafe combinations and apply suppressions to the table and then save the resulting SDC-safe output table. The following sequence of actions is recommended.

* A clean-unit microdata file is prepared (Section 5.1.2)
* A corresponding Tau-Argus metadata file is prepared (Section 5.1.3). Note: this may be produced via the Metadata interface in the program – but a manual preparation is recommended, in particular for complex file structures.
* The file is read into Tau-Argus (Section 5.1.4).
* The output tables to be examined are specified (Section 5.1.5).
* Primary suppression can then be carried out, using either the frequency, magnitude or p% rule. (Section 5.1.5)
* Secondary suppression is then carried out.
* The output can be audited to see if any unsafe combinations remain. Auditing will consider both exact and partial disclosure risk.
* The output table is prepared.

### 5.1.2 Microdata file preparation

Generally, a clean unit microdata file is prepared as standard (as in SAS) then exported as a text or .csv file. Removing the column titles is recommended. Files prepared for processing in uArgus by being stored in one of the following two text-based file formats:

* Comma Delimited File (.csv format)
* Fixed line formats (where columns are specified by character number on each line). For example, Age variable may cover characters 1-3 etc.

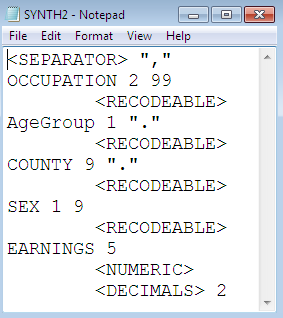
Tau-Argus associates input microdata files with a .ASC file extension. However, it’s not necessary to apply this extension to the input microdata file that has been created. There are advantages to using this extension since it clearly labels the input microdata file.

Changing the extension on the file to “.ASC” can be carried out using Windows or the Command Prompt “ren” command.

### 5.1.3 Metadata file preparation

In addition to the microdata file, Tau-Argus requires accompanying metadata. This is usually prepared in a text editor such as Notepad. It is important to prepare the metadata file correctly, since Tau-Argus will not work correctly. Fig. 2 shows the metadata file for the synthetic dataset.

This is a particularly important part of the process, since the successful use of uArgus is very much dependent on the accurate construction of the metadata file. Fig. 4 shows the structure of the metadata file.



**Fig. 2 Example of metadata file.**

Among the common syntax elements of the metadata file are:

*<SEPARATOR>* specifies that the file is a .csv file.

*<RECODABLE>* specifies that the variable is a categorical variable

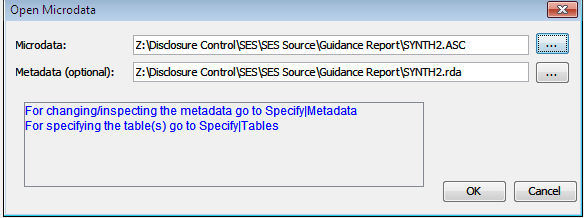
<*NUMERIC*> means that the variable is numeric

<*DECIMALS*> Specifies how many decimal places the numeric variable has.

The value immediately following the column name (e,g OCCUPATION 2 99) have the following meanings: the first number specifies the variable size (i.e. how many characters in the column) while the second number specifies the missing value for the dataset.

### 5.1.4 Reading the file into Tau-Argus

Tau Argus can be applied either to tables or microdata (from which it can generate the tables of interest). In this case, we are examining a microdata file. The microdata file (Synth2.ASC) is loaded via the command “Open Microdata”, with the corresponding metadata file (Synth2.RDA) (Fig 4.):



**Fig. 3 Representation of synthetic dataset being loaded into Tau-Argus**

The next action is to examine the potential output tables of interest to be generated from this microdata file via the “Specify<Tables” menu option. Once this option is selected and the “Specify Tables“ dialog box appears, the microdata file has been loaded into Tau-Argus. However, if there are errors in the metadata file, they usually become apparent in this stage.

### 5.1.5 Preparing tables for Primary Suppression.

We wish to examine the following table for confidentiality:

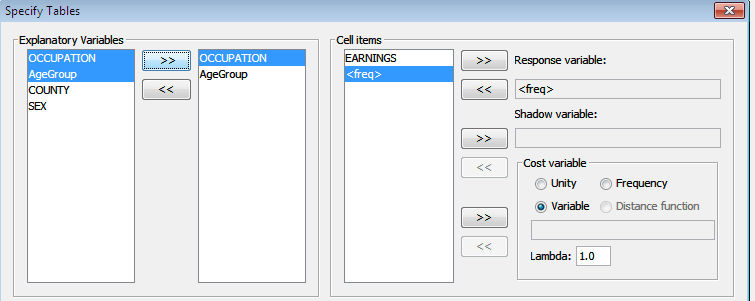
* A frequency table of persons by occupation vs. age group
* A magnitude table (variable earnings) of occupation vs. age group.

We are considering two corresponding confidentiality rules for primary suppression.

* For frequency table, cells are unsafe if there are less than 5 respondents in each cell.
* For the magnitude table, the 2,80% dominance rule (see Section 3.1.1) is being applied here as a primary suppression rule. The variable being considered is Earnings aggregated by Occupation, Sex and County. The application of the dominance rule in this case would mean that any primary cell where two of the respondents contribute more than 80% of the aggregate earnings should be suppressed.

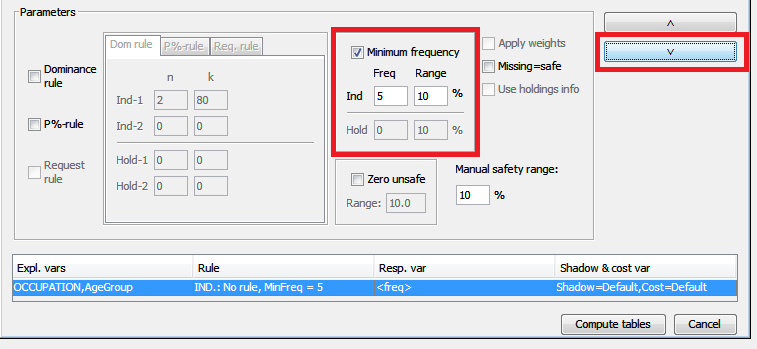
**Frequency tabulation case.**

Firstly, in the Specify Tables dialog box the explanatory variables of interest are selected – occupation and age group. Next, in the “cell items” area of the Specify Tables the <freq> option is specified for the response variable. This tells Tau-Argus that a frequency tabulation is being produced. (Fig. 4),.



**Fig. 4 Specification of explanatory, response variables – frequency tabulation case**

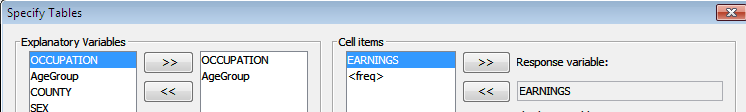
Next, in the Specify Tables dialog box, the primary suppression rule for frequency tables (the minimum frequency rule is specified). The marked button is pressed to indicate that the table is to be generated using the specified rule. “Compute Tables” generates the table.

****

**Fig 5. Applying the minimum frequency rule to the table of interest.**

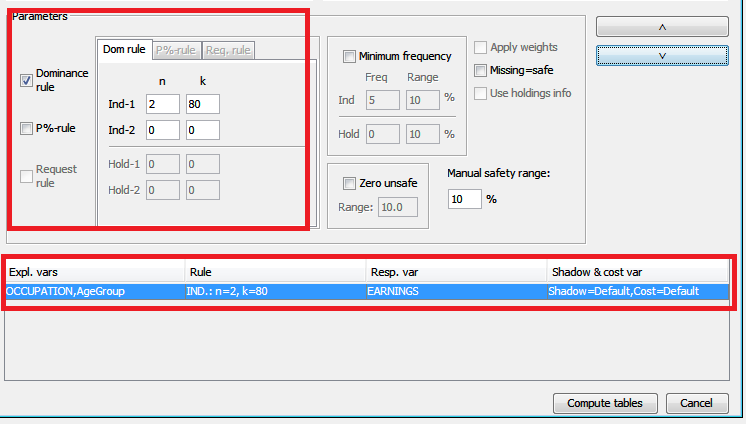
When preparing the second table (the magnitude table), we would apply the following:

* The variable “earnings” would be specified as the response variable rather than <freq>. (Fig. 6)



**Fig. 6 Illustration of “Earnings” variable being specified”.**

* The “Dominance Rule” checkbox would be selected – note that this can be applied at the same time as the “Minimum frequency” rule if required – and the values of 2 and 80 would be entered (Fig. 7).

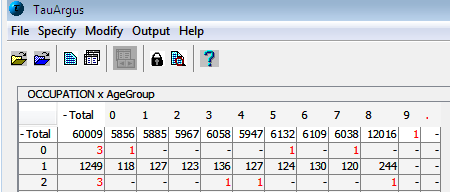


**Fig 7. Illustration of the (2,80) Dominance rule being implemented in Tau-Argus.**

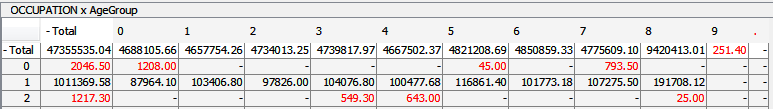
The term “Shadow variable” (Fig. 4) refers to the variable which is being used to specify sensitivity. It is usually left blank, indicating that the response variable is the sensitive one. The “Cost variable” specifies the variable used in guiding the secondary disclosure process (in terms of minimising information loss) – by default it is also the response variable (Refer to Methodology if considering alternative options). “Missing = Safe” option means that missing values are assumed to be safe (this is not recommended).

### 5.1.6 Applying secondary suppression

Tau Argus has now generated the two tables of interest. It has marked in red the cells that are sensitive/disclosive based on the primary suppression rules. Some of these unsafe cells are shown (Fig. 8 and Fig. 9). Secondary suppression is now required.

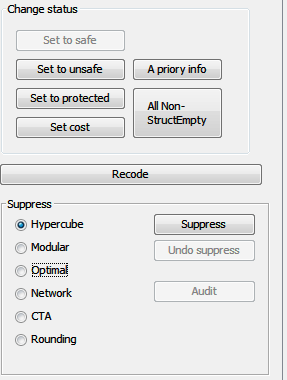


**Fig. 8 Output table – frequency example with unsafe cells**



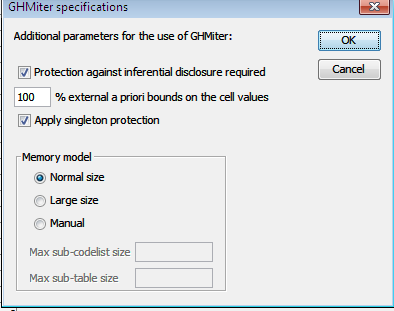
**Fig 9. Output table – magnitude example (Earnings for Occupation x Age Group).**

Next, secondary suppression is applied to each of these tables. The Hypercube, Optimal and Modular methods can be used here. Actions for treating specific cells are also available.

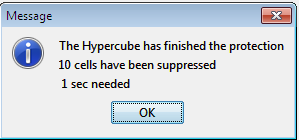


**Fig. 10 Cell Status and Secondary Suppression Options in Tau-Argus.**

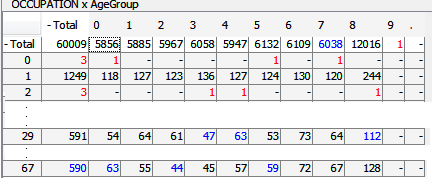
We consider the Hypercube option, as implemented in Tau Argus using the open source GHMiter method, and apply it to the primary suppressed magnitude table in Fig. 9. It is recommended that the standard options be used in the GHMITER Specifications Dialog Box (Fig. 10) which is brought up when the Hypercube option is selected. Note that Hypercube may not run correctly for very large tables. Once the process has run successfully, the dialog box in Fig. 12 is generated. We now see that numerous cells have been suppressed via secondary suppression and marked in blue (Fig. 13)



**Fig. 11 GHMiter specifications for running Hypercube.**

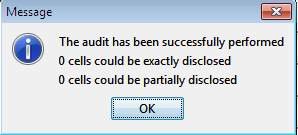


**Fig. 12 Dialog box indicating that the hypercube process has been completed.**



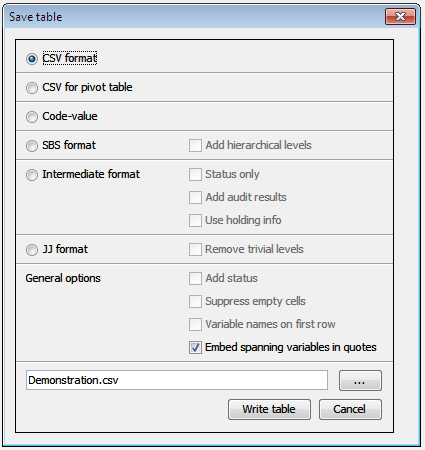
**Fig. 13 Examples of suppressed records marked in blue.**

A significant advantage of Tau-Argus is the ability to audit/examine the results of the secondary suppression to determine if the results are safe. The next step is to run the “audit” function on the data. This ensures that the resulting output is safe and that there is now no possibility of exact or partial disclosure. The “Audit” button in Fig. 10 is selected. Auditing generates two items of output, an extensive report (which can be exported) and a simple summary of the auditing process. Fig 14 shows the summary of the audit process. Ideally, the audit process should indicate that neither exact or partial disclosure is possible. However, if there is a different result, then it is more important that there is zero cells that can be exactly disclosed, since there is a significant risk of primary disclosure otherwise.

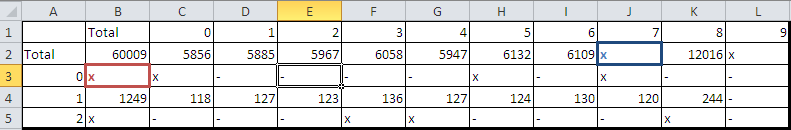


**Fig. 14 Demonstration of successful auditing of the synthetic dataset.**

The menu option >Output>Save Table can now produce a version of this aggregate table including both primary and secondary suppressions (Fig. 15), and the resulting Safe table can be viewed (Fig. 16). It is often simplest to save tables as .csv. The cell highlighted in red is an example of a primary suppressed cell, while the blue cell is a secondary suppressed one. An output report is also generated.



**Fig 15. Save Table Dialog Box**



**Fig. 16 Illustration of suppressed (safe) output table generated by Hypercube process.**

If the user wishes to examine various options, then they can simply select the “Undo suppress” button in Fig. 10 after running each method and then select an alternative method.

## 5.2 Demonstration of primary and secondary suppression using R

For analysing R, the same dataset as in section 5.1 was examined. Primary and secondary suppression were then carried out.

Here, we wish to examine if tables produced from this dataset may require primary and secondary suppression, and, if so, apply these operations via Tau-Argus. The following steps are carried out.

### 5.2.1 R File Structures

R sdcTable requires that the tabular structure of the dataset, including any hierarchical relationships is specified. sdcTable is somewhat rigid in that even for datasets such as SYNTH2 where there are no hierarchical relationships, an overall “Total” column must be created. So, for each of the four categorical variables, a .csv file was created containing the necessary variable information. Table 9 illustrates the structure of the file – no header and no character/string markings, as these are added automatically in R. The single @ sign indicates the highest level of the hierarchy. This logic is easily extended to model hierarchical behaviour.

|  |  |
| --- | --- |
| @ | Total |
| @@ | 9 |
| @@ | 1 |
| @@ | 5 |

**Table 14 Example of how a particular CSV file was prepared.**

**Note on errors**: If there are difficulties with file preparation, these are usually indicated by the following R error messages:

Error in init.dimVar(input = list(input = dimList[[i]], vName = names(dimList)[i])) : "@" must be listed in first row and first column in input!

- This means that the header file includes ‘’ or other unwanted characters.

Error in c\_standardize(inputDims[[i]], input = rawData[[dimVarInd[i]]]) :

c\_standardize: elements of 'codesOrig' not listed in 'codesOriginal' or 'dups'!

- This means that the variables in the metadata files are not matching those in the source file - SYNTH2 in this case. The main cause of this error is if column values in the metadata file do not correspond to those in the microdata files.

### 5.2.2 Data is read into R

In the next step, the SYNTH2 file and the associated metadata files for each variable are read in as R data frames. The metadata files are assigned column names of ‘levels’ and ‘codes’ as a pre-requisite step for R sdcTable. For example, if we consider AgeGroup:

dim.AgeGroup<- read.csv("Z:/Disclosure Control/SES/SES Source/Guidance Report/R work/AgeGroup.csv",header = FALSE, stringsAsFactors=FALSE)

names(dim.AgeGroup)<-c('levels','codes')

This process is repeated for each of the four variable metadata files.

### 5.2.3 Primary suppression in R

Next, the function makeProblem is prepared (Table 10). This analyses the potential combinations of tables and identifies unsafe ones for primary (and latterly secondary) suppression. As a preliminary stage, the variable metadata is combined into a single list called dimList:

dimList <- list(dim.Occupation, dim.AgeGroup, dim.County, dim.Sex)

names(dimList) <- c('Occupation', 'AgeGroup','County','Sex')

Next, the number of categorical variables is defined:

dimVarInd <- c(1,2,3,4)

And the numerical variable (earnings) is specified:

numVarInd <- 5

This demonstration does not use sampling weights or frequency indicator columns so these are marked as null.

freqVarInd <- weightInd <- sampWeightInd <- NULL

The file for SDC analysis is now prepared.

problem <- makeProblem(data=microData1, dimList=dimList, dimVarInd =dimVarInd, freqVarInd=freqVarInd, numVarInd=numVarInd, weightInd= weightInd, sampWeightInd=sampWeightInd).

This creates a tabulation of all possible combinations of tables. This is different to the Tau-Argus approach, where the user can specify the table combinations of interest.

Next, the primary suppression is applied. In this case, we are going to apply primary suppression based on a frequency rule of less than 5 is applied:

problem\_supp<-primarySuppression(problem,type = 'freq',maxN = 5)

And the output is generated (Table 10):

**Table 15 – Primary suppression in R using frequency rule**

Current suppression pattern:

- Primary suppressions: 16594

- Secondary suppressions: 0

- Publishable cells: 24788

It should be noted that there is a much greater number of suppressed cells than in Tau-Argus. The reason for this is that sdcTable is considering a much greater number of possible table combinations compared with our example in Section 5.1, where we simply considered one cross tabulation.

### 5.2.3 Secondary suppression in R

Next step is to apply the Secondary Suppression method to the data. The options available are Hypercube, Optimal (called Opt), Modular (called HITAS) and “SIMPLEHEURISTIC” (the in-house Statistics Austria algorithm for countering exact disclosure). The preferred option here is SIMPLEHEURISTIC since it is more supported by the authors of sdcTable and works extremely well even for very large datasets. While the method only prevents exact “secondary” disclosure, this is not a significant issue in many cases since the risk of inferring “approximate disclosure” can be addressed at the primary suppression stage.

This option is invoked as follows:

resSIMP<-protectTable(problem\_supp, method="SIMPLEHEURISTIC", verbose = TRUE)

The use of the “Verbose = True” flag is recommended since it means that R generates a detailed summary of the suppression process. The following result is generated:

Finished. Total number of new suppressions: 1078

### 5.2.4 Production of output file

Once secondary suppression has been applied, the output file is extracted using the following code:

finalData <- getInfo(resHyper, type='finalData')

This dataset can then be exported as a CSV file.

## 5.3 Using the Visual Basic approach

### 5.3.1 The script

The script for implementing the “nearest neighbour” approach to Primary and Secondary suppression using Visual Basic is relatively simple. It is shown here:

**Sub Stat\_Clear()**

**Dim c As Range**

**For Each c In ActiveSheet.UsedRange**

**If c.Value < 5 And IsEmpty(c.Value) = False Then**

**c.Offset(0, 0).Value = "P"**

**c.Offset(0, 1).Value = "S"**

**c.Offset(1, 1).Value = "S"**

**c.Offset(1, 0).Value = "S"**

**End If**

**Next c**

**End Sub**

Fig. 16 Visual Basic code for primary and secondary suppression.

## 5.3.2 Operation of Script

The code works as follows:

* It runs on active spreadsheet
* Searches for all numbers of less than the c.Value (i.e. the threshold) of 5 (this can be customised).
* Applies primary suppression to this value
* It then applies secondary suppression by clearing a pattern of neighbouring cells (you can customise this - though wouldn't recommend clearing less offset cells).
* This type of secondary suppression is called **“nearest neighbour”.**
* In the above example, three neighbouring cells are suppressed – these locations are specified using the c.Offset command, where the numbers specify the location of the cell or its neighbours.
* However, there is nothing to prevent users from adding in additional suppression of neighbouring cells.
* Unfortunately, it doesn't distinguish between numbers as percentages - as demonstrated in the sheet.
* If you wish to examine its effect on (synthetic) output data, just open the third tab on the attached sheet.
* It will search all tables (in fact numerical values) in Excel

If applied to formula cells (rather than numerical values), it will give a run-time error. Just select output in question and "paste as values" in Excel.

# 6. Recommendations

## 6.1 Recommendations

The following recommendations are made:

1. The final decision on what constitutes acceptable levels of disclosure risk remains with data custodians. The rationale for SDC decisions within business areas must be recorded and retained.
2. There are numerous software solutions for implementing tabular-SDC, and custodians may freely choose which to use. The criteria of a successful solution is whether it, in the informed opinion of the data custodians, reduces disclosure risk to acceptable levels.
3. While final decisions on tabular-SDC belong to data custodians, the SDC process can be assigned to designated staff within a section or even division.
4. A successful SDC implementation in an area may in some, but possibly not all cases, reduce the burden on individual sections, since the risky manual output checking process will hopefully be streamlined or replaced by an appropriate software implementation. None of this is guaranteed however.
5. The ISS, as a whole, needs to be aware of the issues relating to tabular-SDC.

# 7. Further reading

*European Commission.* (2011, September 28). Retrieved November 24, 2016, from http://ec.europa.eu/eurostat/web/products-manuals-and-guidelines/-/KS-32-11-955

Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Lenz, R., Naylor, J., et al. (2010). *Handbook on Statistical Disclosure Control.* ESSNet.

Repsilber, R. (1994). Preservation of Confidentiality in Aggregated data. *Second International Seminar on Statistical Confidentiality.* Luxembourg.

*sdcTable Vignette, Bernhard Meindl, Statistics Austria, 2017*

*τ-ARGUS Version 4.1 User’s Manual, Statistics Netherlands, 2014*

Willenborg, L., & De Waal, T. (1995). *Statistical Disclosure Control in Practice.* New York: Springer.

1. **The authors wish to acknowledge the assistance of Mary Smyth, Cathal Doherty, Barry O’ Leary, Sharon Doyle, Alan Corcoran and Anthony Macken in relation to this project.**  [↑](#footnote-ref-1)
2. Section 34 of the Statistics Act states that “The Office may provide, for statistical purposes only, information obtained in any way under this Act or the repealed enactments, in such form that it cannot be directly or indirectly related to an identifiable person or undertaking…..” [↑](#footnote-ref-2)