Final report WP1 - Multisource editing and analysis

Eurostat grant SMP-ESS-2023-EBS-IBA; 2023-NL-EBS

Sander Scholtus, Jeffrey Hoogland, Frank Aelen, Arnout van Delden, Rob Willems

Statistics Netherlands, 27 June 2025

Index

1. Introduction

1.1 Aim of work package 1 – Multisource editing and analysis

In work package 1 (WP1), on multisource editing and analysis, we aim to develop a system for simultaneous editing of variables that are observed across a collection of different business statistics. In this way we locate and correct large inconsistencies between statistics early in the processing chain. Where possible, we do this automatically, based on rules and reliability weights. In the work package we expand on our earlier work on automatic editing, to make these rules and reliability weights more specific to individual units. In addition, when automatic editing is not feasible or introduces large corrections, a top-down dashboard containing score functions is used to efficiently drill down to large, influential inconsistencies that have to be edited manually. This dashboard, which we developed earlier in a pilot study at Statistics Netherlands, will in this work package be made more generic to deal with various types of data sources, units and output aggregates.

In chapter 2, we provide an overview of our results on automatic data editing and in chapter 3 on top-down editing.

1.2 Short history and outlook of data editing

In the past decades a number of changes have been introduced in the way European business statistics are edited. Traditionally, most units in surveys were edited manually and survey-by-survey. It has been estimated that statistical institutes would spend up to 40% of total business survey costs on editing (Granquist, 1995; Granquist and Kovar, 1997). Manual editing was made more efficient by introducing a number of improvements. First of all, part of the errors were edited automatically, such as obvious errors and errors related to logical rules. Second, score functions were introduced that make use of a difference between the actual and the expected value and also use the contribution of the unit to the total. This way, selective or top-down editing was made possible, where the most influential units are edited manually whereas the others could be edited automatically. A typical pipeline of editing in business statistics is given by Pannekoek et al. (2013).

Another form of editing that has given substantial efficiency gains is macro editing: when aggregates are compared with an expected value and only those aggregates that are suspicious are manually analyzed and edited. An overview of different forms of editing and their relations can also be found in the editing and imputation topics of the Memobust handbook which was developed in an ESSnet project (Scholtus and Willenborg, 2014; Memobust, 2014b). Furthermore, a series of R packages (`validate`, `dcmodify`, `errorlocate`, `simputation`, `rspa`) have been developed to support data editing and imputation (Van der Loo and De Jonge, 2021).

The development and adoption of methods and techniques for data editing and data validation for use within the European Statistical System has been supported by a number of past European projects, including EUREDIT (2000-2003), EDIMBUS (2006-2007), ESSnet ValiDat Foundation (2014-2015) and ESSnet ValiDat Integration (2017-2018).

A more recent development is that NSIs gradually move away from the traditional stovepipe way of producing statistics to a more integrated approach. One of the processing steps in which business statistics are becoming more integrated is that of editing and imputation. That leads to a further improvement of efficiency of data editing and also to an increased quality of the outcomes. A first international development in this context has been that different European NSIs (Denmark, Finland, France, Hungary, Ireland, Italy, Netherlands and Sweden; see Hussain et al. (2018)) started a Large Cases Unit that is responsible for editing and imputation of the largest and most complex units *across different business statistics*. Data editing across statistics for all units, including those not covered by the Large Cases Unit, is a natural extension of this development. However, top-down and automatic editing are necessary to make this feasible and efficient. We expect that our results can also stimulate other European NSIs to profit from our experience and apply automatic and top-down data editing with multiple data sources in the near future.

1.3 References

L. Granquist (1995), Improving the Traditional Editing Process. In: B. G. Cox, D. A. Binder, B. N. Chinnappa, A. Christianson, M. J. Colledge, and P. S. Kott (eds.), *Business Survey Methods*, John Wiley & Sons, New York, pp. 385–401.

L. Granquist and J. Kovar (1997), Editing of Survey Data: How Much Is Enough? In: L. E. Lyberg, P. Biemer, M. Collins, E. D. De Leeuw, C. Dippo, N. Schwartz, and D. Trewin (eds.), *Survey Measurement and Process Quality*, John Wiley & Sons, New York, pp. 415–435.

M. Hussain, R. Peltola and S. Mahajan (2018), Measuring Activities of Multinational Enterprise Groups via Large Cases Units. *Eurostat Review on National Accounts and Macroeconomic Indicators (EURONA)* **1**, 43–58.

Memobust (2014b), Theme: Statistical Data Editing – Main Module. In: *Handbook on Methodology of Modern Business Statistics*, Eurostat.

J. Pannekoek, S. Scholtus and M. van der Loo (2013), Automated and Manual Data Editing: A View on Process Design and Methodology. *Journal of Official Statistics* **29**, 511–537.

S. Scholtus and L. Willenborg (2014), Editing and Imputation in the Memobust Handbook on Methodology of Modern Business Statistics. UNECE Work Session on Statistical Data Editing, 28-30 April 2014, Paris.

M. van der Loo and E. de Jonge (2021), Data Validation Infrastructure for R. *Journal of Statistical Software* **97** (10), 1–31.

2. Automatic editing

To give some context for the activities in the automatic multisource editing part of WP1, we start by a brief introduction on automatic editing (Section 2.1) followed by a brief introduction on multisource editing (Section 2.2). A small example that illustrates the proposed procedure for automatic multisource editing can be found in Appendix A. Section 2.3 describes the data that were used in this study. In addition, manually-edited data were collected for a random subsample, which is described in Section 2.4. Section 2.5 provides a detailed overview of the main methodological developments made during this project, with concrete suggestions for implementing the generic multisource automatic editing procedure from Section 2.2 in practice. Section 2.6 evaluates the outcome of multisource automatic editing for the data in this study, under different scenarios. Finally, details on the implementation of the method in R are given in Section 2.7 and some concluding remarks follow in Section 2.8.

In summary, we have done the following activities:

- Set up a five-step procedure for multisource editing (Section 2.2).
- Carried out a pilot study to obtain a set of data sources with common variables (Section 2.3).
- Prepared a sample for CBS experts from production for manual editing of the data (Section 2.4.1).
- Derived edit rules and correction rules for common variables (Sections 2.4.2, 2.5.1, 2.5.2).
- Used more refined reliability weights (Section 2.5.3).
- Imputed values that were erroneous (Section 2.5.4).
- Calculated different scenarios to assess the individual effect of methods and correction rules (Section 2.5.5).
- Used existing and new measures to evaluate multisource editing (Section 2.6).
- Implemented the five-step procedure in R (Section 2.7).
- Lined up aspects that could still be improved (Section 2.8).

As mentioned in the Grant, activities of the project were presented at the UNECE Expert Meeting on Statistical Data Editing in Vienna, 7-9 October 2024; see Scholtus et al. (2024). Earlier versions of some parts of Sections 2.1, 2.2, 2.4 and 2.5 can also be found in that conference paper. Compared to the conference paper we have made further adjustments and added new parts to make clear what we have done during the rest of the project. In addition, a short presentation on WP1 was given at a meeting of the Working Group on Structural Business Statistics and Business Demography in Luxembourg, 20-21 May 2025.

2.1 Automatic editing methods

Before turning to multisource editing, we will give a brief overview of existing methods for automatic editing of a single data source. Two main classes of methods that are currently in use for automatic editing of business statistics are: deductive correction and error localization based on the Fellegi-Holt paradigm. Deductive correction is intended for systematic errors with a known cause. In practice, deductive correction methods often make use of IF-THEN rules, where the IF condition describes a particular error pattern in the observed data and the THEN condition describes how this error should be corrected. An advantage of deductive correction is that a user has control over the adjustments made to the data in a way that is direct and intuitive. An important disadvantage in some applications is that a large set of IF-THEN rules may be needed to account for all possible error patterns, in which case it tends to become difficult to design and maintain such a set of correction rules (Chen et al., 2003).

Error localization is used to find errors without an obvious cause. For this approach, a user specifies restrictions that should be satisfied by error-free data, known as *edit rules*. Let $\mathbf{x} = \left(x_1, \dots, x_J\right)'$ denote a vector of observed variables. In this paper, we will assume that all variables are real-valued and all edit rules can be written in the following form:

$$\text{IF } (\mathbf{a}_1'\mathbf{x} \le b_1 \text{ AND } \mathbf{a}_2'\mathbf{x} \le b_2 \cdots \text{AND} \cdots \ \mathbf{a}_{K-1}'\mathbf{x} \le b_{K-1}) \text{ THEN } (\mathbf{a}_K'\mathbf{x} \le b_K) \tag{1}$$

for certain known vectors of constants $\mathbf{a}_1, \dots, \mathbf{a}_K$ and constants $b_1, \dots, b_K$. Here, the IF condition may be empty and each $\le$ may also be replaced by $\ge$, $<$, $>$ or $=$. A special case of an edit rule of the form (1) is a simple linear inequality $\mathbf{a}'\mathbf{x} \le b$ or equality $\mathbf{a}'\mathbf{x} = b$. [Note: Error localization methods have also been developed for other types of data, including a combination of categorical and real-valued variables, but we will not treat this topic here; see, e.g., De Waal et al. (2011, Chapters 3-5) and Van der Loo and De Jonge (2018, Chapter 7).]

According to the paradigm of Fellegi and Holt (1976), the error localization problem should be solved by finding the smallest possible subset of variables in $\mathbf{x}$ such that all edit rules can be satisfied by adjusting only these variables. In practice, a generalization of this paradigm is often used, where each variable x_j is given a positive *reliability weight* w_j and the goal is to minimize the sum of the reliability weights of the adjusted variables. Thus, larger reliability weights should be assigned to variables that are less likely to be erroneous. Mathematically, this error localization problem can be written as a mixed-integer linear programming (MILP) problem (Van der Loo and De Jonge, 2018):

$$\begin{array}{l} \min\left(\sum_{j=1}^{J} w_j \delta_j\right) \text{ under the following restrictions:} \\ \tilde{\boldsymbol{x}} = \left(\tilde{x}_1, \dots, \tilde{x}_J\right)' \text{ satisfies all edit rules of the form (1);} \\ x_j - M\delta_j \le \tilde{x}_j \le x_j + M\delta_j \text{ for all } j \in \{1, \dots, J\}; \\ \boldsymbol{\delta} = \left(\delta_1, \dots, \delta_J\right)' \in \{0,1\}^J. \end{array} \tag{2}$$

Here, δ_j is a binary variable that indicates whether variable x_j is to be adjusted ($\delta_j = 1$) or not ($\delta_j = 0$), and $\tilde{\boldsymbol{x}}$ denotes the adjusted record. In addition, M is a large positive number which should be chosen an order of magnitude larger than any value that is expected in $\mathbf{x}$. Note that the restriction in the third line of (2) implies that $\tilde{x}_j = x_j$ when $\delta_j = 0$ (i.e., the original value is not adjusted) and $-M \le \tilde{x}_j - x_j \le M$ when $\delta_j = 1$ (i.e., in practical terms any adjustment can be made to the original value). In this study we have used $M = 10^7$. If the original record $\mathbf{x}$ contains any missing values, then these may be imputed 'for free' by any value; the corresponding values of $\tilde{\boldsymbol{x}}$ are unrestricted in (2). It should be noted that the computation time needed to solve the MILP problem can become impractically large if many conditional edit rules of the form (1) are included; purely linear edit rules are handled much more efficiently.

A solution to error localization problem (2) consists of only the error pattern $\boldsymbol{\delta}$. In general, there may exist an infinite number of possible adjusted records $\tilde{\boldsymbol{x}}$ that satisfy the restrictions in (2) for a given solution $\boldsymbol{\delta}$. In practice, the final adjusted record can be created by, first, setting the erroneous values to missing, second, imputing new values for all variables with missing values and, third, adjusting only these imputed values if necessary so that all edit rules (1) become satisfied. This last step can be formulated as a linear or quadratic programming problem. In practice, solving such a problem is much less computationally demanding than solving the MILP problem (2). See, e.g., De Waal et al. (2011, Chapter 10) for more details.

In general, the optimal solution of the error localization problem is determined by the choice of reliability weights. Liepins (1980) showed that an optimal solution to problem (2) can be seen as an approximate maximum likelihood estimator of the true error pattern under a particular model for random measurement errors, provided that the reliability weights are chosen as $w_j = \log(1 - p_j) - \log(p_j)$, where p_j denotes the probability that an error has occurred in x_j. In principle, these 'optimal' weights could differ for each unit. In practice, reliability weights are usually determined by other methods than by predicting the probabilities $p_1, \dots, p_J$; we will return to this issue in Section 2.5.3.

In the above error localization problem, it was assumed that all edit rules (1) are hard edit rules (i.e., they must be satisfied by any error-free record). In practice, soft edit rules can also occur which indicate situations that are implausible but not impossible. Scholtus (2015) proposed an extension of MILP problem (2) that can accommodate soft edit rules. For each soft edit rule, a positive weight s_m is assigned and an auxiliary variable $\zeta_m \in \{0,1\}$ is introduced which indicates whether soft edit rule m may be violated ($\zeta_m = 1$) or not ($\zeta_m = 0$). Each soft edit rule of the form (1) is now replaced by a hard edit rule of the form

$$\text{IF } (\zeta_m = 0 \text{ AND } \mathbf{a}_1'\mathbf{x} \le b_1 \text{ AND } \mathbf{a}_2'\mathbf{x} \le b_2 \cdots \text{AND} \cdots \mathbf{a}_{K-1}'\mathbf{x} \le b_{K-1}) \text{ THEN } (\mathbf{a}_K'\mathbf{x} \le b_K).$$

The target function in (2) is extended to $\sum_{j=1}^{J} w_j \delta_j + \sum_{m=1}^{M} s_m \zeta_m$. In the initial data, all auxiliary variables $\zeta_1, \dots, \zeta_M$ are filled with zeros. Thus, each soft edit rule that remains violated in a proposed solution to the error localization problem will contribute its weight s_m to the target function of the optimization problem, as ζ_m has to be changed from 0 to 1 to accommodate this edit violation. The optimal solution now involves a balance between the sum of reliability weights of any variables that are edited and the sum of weights of any soft edit rules that remain violated. Note that this extended problem can be solved by any tool that could solve the original problem.

Another limitation of Fellegi-Holt-based error localization is that it is based on the assumption that errors occur independently in each variable. However, sometimes errors occur for which it is natural to correct them by adjusting multiple variables at once. For instance, a respondent could interchange the values of two variables by mistake. Daalmans and Scholtus (2018) formulated an extension of error localization problem (2) that can accommodate a more general class of *edit operations*, including operations that affect more than one variable. They showed that this extended error localization problem can also be solved as a MILP problem.

Finally, it should be noted that other automatic editing methods have been developed. For instance, Little and Smith (1987) proposed an editing method based on an explicit statistical model for the data and Dumpert (2020) and Rocci (2020) discuss some recent proposals to use machine learning for editing. Many of these other methods implicitly use soft edit rules but do not easily incorporate hard edit rules. One exception is the Nearest-neighbour Imputation Methodology developed by Statistics Canada for the household census (Bankier, 2006; De Waal et al., 2011, Section 4.5).

2.2 Multisource editing

2.2.1 Notation and setup

Denote the observed source-specific variables for unit i in data source $p \in \{1, \dots, P\}$ by $\mathbf{y}_i^{(p)} = \left(y_{i1}^{(p)}, \dots, y_{iJ_p}^{(p)}\right)'$. Within each data source, there may be internal edit rules of the form (1) that should be satisfied:

$$\text{IF } \left(\mathbf{a}_1'\mathbf{y}_i^{(p)} \le b_1 \text{ AND } \mathbf{a}_2'\mathbf{y}_i^{(p)} \le b_2 \cdots \text{AND} \cdots \mathbf{a}_{K-1}'\mathbf{y}_i^{(p)} \le b_{K-1}\right) \text{ THEN } \left(\mathbf{a}_K'\mathbf{y}_i^{(p)} \le b_K\right). \quad (3)$$

A *common variable* is a variable that occurs in at least two data sources, with definitions that are aligned so it is reasonable to expect the same unit to report the same value in each source. Suppose that across all data sources we have identified L common variables. Typically, $L \ll \sum_{p=1}^{P} J_p$. Let $x_{il}^{(p)}$ denote the value of common variable l for unit i in data source p. In practice, each data source will contain only a subset of all common variables. Let $\mathbf{x}_i^{(p)}$ denote a vector containing all $x_{il}^{(p)}$ that occur in data source p. In general, a common variable may not be directly observed in a source but has to be derived from the source-specific variables $\mathbf{y}_i^{(p)}$. We assume here that all derivations are affine transformations, so it holds that

$$\mathbf{x}_{\mathrm{i}}^{(p)} = \mathbf{C}^{(p)}\mathbf{y}_i^{(p)} + \mathbf{d}^{(p)} \quad (4)$$

for some known matrix $\mathbf{C}^{(p)}$ and vector $\mathbf{d}^{(p)}$ of constants.

The actual sources that are available for each common variable differ by unit, because of sampling, non-response and other data collection issues. Let B_{il} denote the subset of sources $\{1, \dots, P\}$ in which common variable l is (indirectly) observed for unit i. Since our aim is to avoid large inconsistencies between the observed values of common variables in different data sources, we define further restrictions of the following form:

$$\left|x_{il}^{(p)} - x_{il}^{(q)}\right| \le A_l\left(x_{il}^{(q)}\right), \quad (5)$$

for all pairs (p, q) with $p \in B_{il}$ and $q \in B_{il}$. Here, $A_l(x)$ is a function that defines the maximally allowed deviation between two observed values of common variable l for the same unit. Different choices are possible for this function, which will be discussed in more detail in Section 2.2.2.

Choosing $A_l(x) = 0$ in (5) would mean no deviations are allowed at all. In practice, this choice would require us to resolve many very small inconsistencies. It may be more convenient to leave relatively small inconsistencies unresolved at the level of individual units. Consistent statistical output could then still be obtained by applying techniques such as macro-integration (Mushkudiani et al., 2014) or calibration (Deville and Särndal, 1992) to resolve the remaining inconsistencies at a higher level of aggregation.

To formulate the automatic multisource editing problem, it is useful to introduce a unique vector of values of common variables for unit i, $\mathbf{z}_i = (z_{i1}, \dots, z_{iL})'$. These will be referred to as *universal values* of the common variables. Informally, we can think of $\mathbf{z}_i$ as (our best prediction of) a unit's true values of the common variables. Instead of (5), we may then define restrictions of the form

$$\left|x_{il}^{(p)} - z_{il}\right| \le A_l^*(z_{il}) \quad (6)$$

for all $p \in B_{il}$. The function $A_l^*(z)$ is determined by $A_l(x)$ used in (5); see Section 2.2.2.

In addition, we may define other edit rules of the form (1) for the common variables:

$$\text{IF } (\mathbf{a}_1'\mathbf{z}_i \leq b_1 \text{ AND } \mathbf{a}_2'\mathbf{z}_i \leq b_2 \cdots \text{AND} \cdots \; \mathbf{a}_{K-1}'\mathbf{z}_i \leq b_{K-1}) \text{ THEN } (\mathbf{a}_K'\mathbf{z}_i \leq b_K). \qquad (7)$$

Edit rules of the form (7) may also involve $x_{il}^{(p)}$ but not $y_{ij}^{(p)}$. At the start of the editing process, the universal values z_{il} are unknown (i.e., missing).

In brief, the purpose of automatic multisource editing is to obtain data for unit i, consisting of $\left(\mathbf{x}_i^{(1)}, \dots, \mathbf{x}_i^{(P)}, \mathbf{y}_i^{(1)}, \dots, \mathbf{y}_i^{(P)}, \mathbf{z}_i\right)$, that satisfy all edit rules (3), (4), (6) and (7). Scholtus et al. (2022) discussed that solving this automatic editing problem in one step becomes increasingly challenging as more data sources and common variables are added. Instead, we propose a five-step procedure:

1) <u>Deductive correction of common variables across data sources</u>. Correction rules are applied. These rules correct for systematic errors in $x_{il}^{(p)}$ and $y_{ij}^{(p)}$.
2) <u>Specifying reliability weights of common variables</u>. Starting from a set of fixed reliability weights $w_j^{(p)}$, unit-specific reliability weights $w_{ij}^{(p)}$ are derived by applying additional IF-THEN rules and other techniques that use unit-specific information.
3) <u>Automatic editing of common variables across data sources</u>. Errors are identified in the common variables $\left(\mathbf{x}_i^{(1)}, \dots, \mathbf{x}_i^{(P)}, \mathbf{z}_i\right)$, using edit rules (6) and (7).
4) <u>Imputing universal values and deriving additional edit rules for the common variables</u>. The universal values $\mathbf{z}_i$ are imputed in line with the edit rules (6) and (7). These imputed values are substituted in (6) to obtain a set of edit rules for $\left(\mathbf{x}_i^{(1)}, \dots, \mathbf{x}_i^{(P)}\right)$.
5) <u>Automatic editing within each individual data source</u>. This step is carried out independently for each data source. Errors are identified in the observed values $\left(\mathbf{x}_i^{(p)}, \mathbf{y}_i^{(p)}\right)$ in each data source separately, using the edit rules (3) and (4), as well as the relevant edit rules from (6). The imputed universal values $\mathbf{z}_i$ from step 4 may not be edited during this step.

In step 5, both deductive correction and error localization [or any other automatic editing method that can account for edit rules of the form (3), (4), (6) and (7)] could be applied. The basic procedure is illustrated using a sample case in Appendix A of this report. A detailed description of the last three steps, as well as a larger, more realistic example, can be found in Scholtus et al. (2022).

In addition, it may be useful to add a step 0 before the five-step procedure, in which obvious systematic errors – such as thousand errors – are corrected deductively in each individual data source before the common variables are derived:

0) <u>Deductive correction of source-specific variables for individual data sources</u>. Correction rules are applied to correct obvious systematic errors that can be found without comparing data across different sources. These rules correct for errors in $y_{ij}^{(p)}$, thereby improving the quality of the initial derived values of the common variables $x_{il}^{(p)}$.

Since step 0 does not involve any multisource editing, we do not consider it as part of the five-step procedure.

Each step of the above procedure still requires some choices to be made: which edit rules to use, which correction rules to use, how to set certain parameters of the methods, etc. The specification we used for the maximally allowed deviation between observed versions of the same common variable is discussed in the next subsection. The other methodological choices made in this study will be described in Section 2.5.

2.2.2 Defining inconsistencies

Scholtus et al. (2022) proposed to define consistency rules (5) by bounding the relative deviation between observed values of the same common variable:

$$\left|x_{il}^{(p)} - x_{il}^{(q)}\right| \leq \varepsilon_l \left|x_{il}^{(q)}\right| \tag{8}$$

for all pairs (p, q) with $p \in B_{il}$ and $q \in B_{il}$. Here, the parameter $0 \leq \varepsilon_l < 1$ defines the maximally allowed relative deviation between observed values of common variable l. For instance, $\varepsilon_l = 0.1$ means that relative deviations of up to 10% are allowed. The corresponding restrictions of the form (6) are given by

$$\left|x_{il}^{(p)} - z_{il}\right| \leq \varepsilon_l^* |z_{il}| \tag{9}$$

for all $p \in B_{il}$. With the choice $\varepsilon_l^* = \varepsilon_l/(2 + \varepsilon_l)$ or, equivalently, $\varepsilon_l = 2\varepsilon_l^*/(1 - \varepsilon_l^*)$, it can be shown using the triangle inequality that any set of values that satisfies all restrictions (9) also satisfies all restrictions (8) (Scholtus et al., 2022).

It turns out that this criterion which only accounts for relative deviations identifies many inconsistencies that are small in absolute terms, in particular for small businesses. Since many of these small inconsistencies are not of interest and difficult to correct, it seems convenient to extend $A_l(x)$ in (5) with a measure of absolute deviation. For each common variable l we define a maximally allowed absolute deviation $\Delta_l^* \geq 0$ from z_l, in addition to the maximally relative deviation ε_l^* that was already defined. A deviation will now be seen as an inconsistency if it is larger than both the allowed relative and absolute margins. A natural way to specify this as a restriction is by replacing (9) by:

$$\left|x_{il}^{(p)} - z_{il}\right| \leq \max\{\varepsilon_l^* |z_{il}|, \Delta_l^*\} \tag{10}$$

for all $p \in B_{il}$.

To use restriction (10) for automatic editing within this study, it has to be replaced by an equivalent set of edit rules of the form (1):

$$\begin{aligned}
&\text{IF } \left(z_{il} \geq 0 \text{ AND } x_{il}^{(p)} < z_{il} - \Delta_l^*\right) \text{ THEN } \left(x_{il}^{(p)} \geq (1 - \varepsilon_l^*) z_{il}\right); \\
&\text{IF } \left(z_{il} \geq 0 \text{ AND } x_{il}^{(p)} > z_{il} + \Delta_l^*\right) \text{ THEN } \left(x_{il}^{(p)} \leq (1 + \varepsilon_l^*) z_{il}\right); \\
&\text{IF } \left(z_{il} < 0 \text{ AND } x_{il}^{(p)} > z_{il} + \Delta_l^*\right) \text{ THEN } \left(x_{il}^{(p)} \leq (1 - \varepsilon_l^*) z_{il}\right); \\
&\text{IF } \left(z_{il} < 0 \text{ AND } x_{il}^{(p)} < z_{il} - \Delta_l^*\right) \text{ THEN } \left(x_{il}^{(p)} \geq (1 + \varepsilon_l^*) z_{il}\right).
\end{aligned} \tag{10*}$$

We tested these rules in practice but found that error localization became too slow for practical purposes, due to the large number of complex conditional edit rules involved in the MILP problem. We therefore decided to approximate (10) instead by an alternative restriction:

$$\left|x_{il}^{(p)} - z_{il}\right| \leq \Delta_l^* + \varepsilon_l^* |z_{il}| \tag{11}$$

for all $p \in B_{il}$. For common variables that are restricted to be non-negative ($z_l \geq 0$) – which amounts to the vast majority of common variables in our study – restriction (11) is equivalent to two linear edit rules:

$$\begin{aligned} x_{il}^{(p)} &\geq (1 - \varepsilon_l^*) z_{il} - \Delta_l^*; \\ x_{il}^{(p)} &\leq (1 + \varepsilon_l^*) z_{il} + \Delta_l^*. \end{aligned} \tag{11*}$$

For common variables that can have both negative and positive values, restriction (11) does require conditional edit rules of the form (1), but these are less complex than (10*):

$$\begin{aligned} &\text{IF } (z_{il} \geq 0) \text{ THEN } \left(x_{il}^{(p)} \geq (1 - \varepsilon_l^*) z_{il} - \Delta_l^* \right); \\ &\text{IF } (z_{il} \geq 0) \text{ THEN } \left(x_{il}^{(p)} \leq (1 + \varepsilon_l^*) z_{il} + \Delta_l^* \right); \\ &\text{IF } (z_{il} < 0) \text{ THEN } \left(x_{il}^{(p)} \geq (1 + \varepsilon_l^*) z_{il} - \Delta_l^* \right); \\ &\text{IF } (z_{il} < 0) \text{ THEN } \left(x_{il}^{(p)} \leq (1 - \varepsilon_l^*) z_{il} + \Delta_l^* \right). \end{aligned} \tag{11**}$$

This alternative formulation of the consistency rules turned out to yield acceptable computation times for error localization in our study.

To find an interpretation of (11) in terms of the observed deviations $\left| x_{il}^{(p)} - x_{il}^{(q)} \right|$, we note that the triangle inequality implies that, for any record that satisfies all restrictions of the form (11),

$$\begin{aligned} \left| x_{il}^{(p)} - x_{il}^{(q)} \right| &\leq \left| x_{il}^{(p)} - z_{il} \right| + \left| x_{il}^{(q)} - z_{il} \right| \\ &\leq 2\Delta_l^* + 2\varepsilon_l^* |z_{il}| \\ &\leq 2\left(1 + \frac{\varepsilon_l^*}{1 - \varepsilon_l^*} \right) \Delta_l^* + 2 \frac{\varepsilon_l^*}{1 - \varepsilon_l^*} \left| x_{il}^{(q)} \right|. \end{aligned}$$

In the last line it was used that $|z_{il}| \leq \left| x_{il}^{(q)} \right| + \left| x_{il}^{(q)} - z_{il} \right| \leq \left| x_{il}^{(q)} \right| + \Delta_l^* + \varepsilon_l^* |z_{il}|$, which implies that $|z_{il}| \leq \left(\left| x_{il}^{(q)} \right| + \Delta_l^* \right) / (1 - \varepsilon_l^*)$. Now with $\varepsilon_l = 2\varepsilon_l^* / (1 - \varepsilon_l^*)$ as defined above, we obtain:

$$\left| x_{il}^{(p)} - x_{il}^{(q)} \right| \leq (2 + \varepsilon_l) \Delta_l^* + \varepsilon_l \left| x_{il}^{(q)} \right|. \tag{12}$$

Thus, deviations between observed values of the same common variable in different sources are roughly bounded by ε_l in relative terms and $\Delta_l = (2 + \varepsilon_l) \Delta_l^* = (\varepsilon_l / \varepsilon_l^*) \Delta_l^*$ in absolute terms.

Note that the right-hand-side of (12) can be rewritten as $\varepsilon_l \left(\Delta_l^* / \varepsilon_l^* + \left| x_{il}^{(q)} \right| \right)$. Therefore, for units with large observed values $\left| x_{il}^{(q)} \right| \gg \Delta_l^* / \varepsilon_l^*$, restriction (12) is determined mainly by the relative deviation: the upper bound is approximately equal to $\varepsilon_l \left| x_{il}^{(q)} \right|$ and (12) reduces to (8). For units with small values $\left| x_{il}^{(q)} \right| \ll \Delta_l^* / \varepsilon_l^*$, the upper bound is approximately equal to Δ_l and restriction (12) is determined mainly by the absolute deviation. Hence, as intended, we expect that restrictions (11) and (12) yield fewer inconsistencies among small businesses than the original formulation based on (8) and (9), while for larger businesses the two formulations are approximately equivalent.

In this study we have used edit rules (11*) and (11**) with the following choice of parameter values: $\varepsilon_l^* = 0.05$ for all common variables, $\Delta_l^* = 2$ for the common variable 'average number of employees in fte (full-time equivalents)' and $\Delta_l^* = 50$ for all other common variables. Note that all common variables besides the number of employees are financial variables expressed in multiples of €1 000, so $\Delta_l^* = 50$ actually represents an absolute deviation of €50 000.

2.3 Pilot studies

In an earlier pilot study before the start of the EBS grant, a prototype implementation of steps 3, 4 and 5 of the five-step procedure for automatic multisource editing from Section 2.2.1 was developed using a suite of existing R packages: `validate` and `validatetools` for managing and evaluating edit rules, `dcmodify` for deductive correction, `errorlocate` for Fellegi-Holt-based error localization, `deductive` and `simputation` for imputation of missing values, and `rspa` for adjusting imputed values to edit rules by quadratic minimization (existing R-packages: Van der Loo and De Jonge, 2018 and 2021). The initial pilot study was conducted in 2021 and 2022 with data from $P = 7$ sources, with $L = 13$ common variables and over 100 variables in total. The main finding of this pilot study was that the multisource editing approach is computationally feasible but that the quality of edited data was not yet good enough for use in actual production (Scholtus et al., 2022). To improve the quality of automatic editing, more subject-matter knowledge should be included in the method.

For the pilot study carried out within this EBS grant, we made some changes compared to the previous study with respect to the data. In the current study, $P = 9$ data sources were used:

- Structural Business Statistics (SBS; survey)
- Investment Statistics (IS; survey)
- ProdCom (survey)
- Statistics on Finances of Large Enterprise groups (SFLE; survey)
- Short-Term Statistics (STSA; admin data)
- Short-Term Statistics (STSS; survey)
- Statistics on Employees and Salaries (SES; admin data)
- Statistics on International Trade of Goods and Services (SIGS; combination of survey and admin data)
- Profit Declaration (PD; admin data)

[Note: Throughout this report, the term “Structural Business Statistics” (SBS) refers to an annual business survey which covers most, but not all, of Statistics Netherlands’ SBS output. In particular, the IS survey contains additional SBS variables.]

In total, $L = 33$ common variables have been identified for this study. Of these, 27 variables occur in exactly two sources, five variables occur in exactly three sources, and one variable (‘net turnover minus excise duties’) occurs in four sources. See Appendix B for an overview of all common variables and the data sources in which they occur. Most common variables have to be derived from source-specific variables by a transformation of the form (4). All data refer to the year 2022. In addition, production-edited data for the year 2021 were available as reference data.

Because the whole set of available NACE codes is very broad, we decided to narrow down the scope for our analyses to a set of seven, diverse, NACE groups. Below we give a description of those NACE groups with a short motivation for their inclusion in the study. Note: the NACE codes and descriptions below are based on the Dutch implementation of the NACE classification.

- *Manufacture of bakery, pastry and farinaceous products* (NACE 107). There is a significant sample overlap across the sources and it is an industry with normal editing issues. Bakeries have different business models (a traditional bakery with own shops and a bread factory with sales to retailers).

- *Construction* (NACE 412). This is a reasonably heterogeneous industry with both smaller and larger enterprises.
- Three groups within *wholesale trade* (NACE 465, 461, 463):
    - *Wholesale on a fee or contract basis* (NACE 461). There are often issues with the ownership of goods. When companies become the owners of the goods, they must be recorded at their gross value. If this is not the case, it is considered a pure service, and only the fee for this service should be recorded as revenue. Therefore inconsistencies across sources are expected.
    - *Wholesale of food and beverages* (NACE 463). We expect relatively many normal enterprises, but different types of wholesalers: importers, exporters, domestic trade, transit trade. Perhaps some enterprises have issues with royalty fees.
    - *Wholesale of information and communication equipment* (NACE 465). Possible issues are the international character of the enterprises. The wholesale activity is often combined with service activities (NACE 62).
- *Freight transport by road* (NACE 494). This is a large group in the transport industry with potential editing issues due to cross-border activities.
- Other transportation support activities (*forwarding agencies, ship brokers and charterers; weighing and measuring*, NACE 5229). Measurement issues can occur in administrative data (unjustified trade flows). These enterprises are often part of an international corporation or they have a lot of activities for foreign enterprises.
- *Restaurants* (NACE 561). This is a large NACE group and we expect to encounter more standard editing issues here.
- *Support activities in the field of information technology* (NACE 620). This is a heterogeneous group which might involve issues with ownership relations. It contains enterprises that produce their own software but also more complex IT companies. Part of the turnover will be non-domestic.

Our starting point is that for the incorporation of subject-matter knowledge as well as for the evaluation of the multisource editing, we focus on those seven groups. Moreover, we focus on enterprises with less than 200 employees that are not part of a larger enterprise group, as larger and more complex units are more likely to be suitable for top-down manual editing. Finally, since in current production at Statistics Netherlands the most advanced form of automatic editing is used for SBS, we focus on SBS respondents. In the 2022 data, 9560 enterprises are observed in two or more sources that satisfy all of these criteria.

In this study, steps 1, 2, 3 and 4 of the multisource automatic editing procedure from Section 2.2 were applied to the above-mentioned sources. Step 5 was applied only to SBS data. SBS is the only statistic considered in this study which already uses error localization based on the Fellegi-Holt paradigm in the current production process at Statistics Netherlands. Hence, this would be the first process at Statistics Netherlands where the complete multisource automatic editing procedure, including step 5, could be implemented in production.

2.4 Manual editing of a sample

A natural way to evaluate the quality of automatic editing is by comparing automatic editing to manual editing, under the assumption that manually edited data are the ‘gold standard’. However, in our case it is difficult to evaluate the quality of automatic multisource editing based on historical manually edited data alone, for at least two reasons. First, manual editing during regular production is reserved for the largest and most complicated cases, so the manually edited data are not a representative sample for the whole population. Second, due to the stovepipe nature of current production processes (as discussed in Section 1.2), manual editing on historical data was often done without taking consistency across different statistics explicitly into account. Only a limited number of CBS experts has a lot of insight in most of the data sources that are used in the pilot study. Therefore, manually edited data may not be considered as a ‘gold standard’ for our purposes.

2.4.1 Drawing the sample

To obtain a better data set for evaluation, we have drawn a probability sample of 350 units from the pilot study data, to be edited manually outside of regular production with the multisource aspect taken into account. For the units in the sample, statistical analysts have been asked to explain all inconsistencies between common variables in the raw data that are larger than 10% and to correct any erroneous values that they find. An R Shiny dashboard was developed for this exercise, where analysts can edit the data and provide comments on their findings. The sample of 350 units was drawn as a stratified sample of 50 units each from the seven different industries of 2022 data, mentioned in Section 2.3. Units without any inconsistencies larger than 10% on common variables or with at least one inconsistency larger than 200% were not eligible for selection: the former do not require multisource editing, the latter may not be suitable for automatic editing. For the same reason, large units with 200 employees or more and enterprises that are part of a larger enterprise group were also excluded.

2.4.2 Results on the sample

The sample was drawn in April 2024 and from then analysts started to edit the raw data. These analysts had varying levels of experience with editing only data from individual statistics. We have discussed the progress with the analysts several times. With the exception of the most experienced analysts, they found it difficult to edit the data over different sources. It requires a lot of knowledge about the different sources used and it is often not clear why the values in the data sources differ. In principle the analysts were allowed to contact the enterprises that have sent in the survey data in order to find out the reasons behind the differences. In practice, the editors were reluctant to contact the enterprises because: 1) they lack a deep understanding and an overview of all the statistics; 2) it concerns 2022 data and it does not feel comfortable to ask questions about historical data, also because in the past it was forbidden to contact enterprises to ask questions about administrative data.

For a few of the industries manual editing went relatively well, namely for construction (NACE 412), wholesale on a fee or contract basis (NACE 461), restaurants (NACE 561), and computer programming (NACE 620). For these sectors analysts were available that had knowledge about issues regarding both surveys and registers.

A group of experts did a second round of manual editing for these four industries. They discussed part of the units together. One of the goals was to increase the number of units for which all violated edit rules are solved. The idea is that all units for which the edit rules for common variables are no longer violated, could be used later to evaluate the performance of the automatic editing, see Section 2.6.1.

We asked the analysts to write down in comments what edit choices they have made and why they made those choices. Those comments were analyzed in order to

- find deductive correction rules for common variables
- improve the initial reliability weights per source and common variable
- derive reliability weight per source, common variable and enterprise by means of deductive rules
- find criteria to assess the completion quality of an SBS-survey

In total, the four industries that were edited in the second round contained 166 sampled units: 16 units from NACE 461 and 50 units each from the other industries. (Recall that 50 units were originally sampled jointly from NACE codes 461, 463, and 465.) For 70 units, the analysts themselves concluded there was at least one edit violation among common variables for which no satisfactory explanation could be found within the time available. Of the remaining 96 units, 14 units had at least one edit violation remaining in the manually edited data. Thus, all edit violations were resolved for 82 units in the sample in these four industries. For these 82 units, the mean number of edit violations among common variables in the original data was 3.24, the median was 2.00, and the maximum was 18.

For the evaluation we restrict attention to NACE 412, 461 561, and 620. We also restrict attention to part of the data sources, because the available analysts and experts did not have knowledge about all data sources. After an analysis of the comments and corrections of the analysts it was concluded that part of the manual corrections of inconsistencies were not correct, or were not properly substantiated. Records regarding these corrections were restricted from the evaluation. Records that still contain large inconsistencies or violations of edit rules regarding SBS, STSS, STSA, SES and PD were also discounted. Furthermore, additional correction rules are only derived if the analyst has a good story. This analysis led to a further reduction compared to the 82 units mentioned above.

The final number of records in the sample that is used for derivation of deductive correction rules and evaluation purposes is 69. Despite the limited number of records it is a very useful data set. The comments of analysts are often extensive and contain useful information to derive deductive rules and update reliability weights. Nevertheless, the comments of analysts would have been even more useful if they had contacted the respondents about inconsistencies. That would also have led to a larger data set for evaluation purposes.

2.5 Methods used within automatic multisource editing

Recall that the automatic editing procedure outlined in Section 2.2 consists of five steps:

1) Deductive correction of common variables across data sources.
2) Specifying reliability weights of common variables.
3) Automatic editing of common variables across data sources.
4) Imputing universal values and deriving additional edit rules for the common variables.
5) Automatic editing within each individual data source.

An important aim of the current project was to develop ways to take more subject-matter knowledge into account during these steps and apply these to the pilot study data. Three types of input parameters that directly affect the outcome of automatic editing are:

(i) edit rules that should be satisfied by the data (these affect all steps, but most directly steps 3, 4 and 5);

(ii) deductive correction rules (applied in step 1);

(iii) reliability weights of common variables (set in step 2 and applied in step 3).

The methodology used in this pilot study to define these three types of input parameters will be discussed in Sections 2.5.1, 2.5.2, and 2.5.3, respectively. Next, the imputation methodology used in step 4 is discussed in Section 2.5.4. Finally, Section 2.5.5 describes different scenarios that were tested in this pilot study, to highlight the contribution of different parts of the methodology.

2.5.1 Finding additional relevant edit rules

As explained earlier in Section 2.2, the proposed multisource error localization problem involves edit rules of the forms (3), (4), (6) and (7). Specifying the first three of these is relatively straightforward. Internal edit rules (3) for most data sources are already well-developed as part of regular statistical production. Exceptions may occur for some sources, e.g., for administrative data that are not yet used directly to create statistical output; in the pilot study this is true for PD. Edit rules (4) relating $\mathbf{y}_i^{(p)}$ to $\mathbf{x}_i^{(p)}$ are given by definition. Edit rules (6) are operationalized in this study by (11*) for non-negative common variables and (11**) for all other common variables, as explained in Section 2.2.2; this requires only a specification of the parameters ε_l^* and Δ_l^*. By contrast, edit rules (7) for the universal values of common variables are still mostly lacking. Thus, finding edit rules for these variables seems to be a good opportunity for improvement.

The lack of explicit edit rules for common variables reflects a wider issue: outside of the Large Cases Unit, statistical analysts currently have little experience with comparing these variables across statistics. While the experiences of the Large Cases Unit are useful and in fact crucial here – it is the main source of the definitions of common variables used in (4) –, it is also limited to the largest and most complicated enterprise groups, whereas automatic editing will be focused mainly on small to medium-sized enterprises without a complicated structure. More knowledge of relations between common variables for these smaller units is therefore needed. In the future, this knowledge should increase naturally over time as top-down interactive multisource editing becomes more widespread as part of regular statistical production. Within this grant, we used a data-driven approach to find relations between common variables that could be turned into edit rules.

As a starting point, historical (internally) edited values $x_{il}^{(p)}$ in one particular data source p were used as a proxy for the underlying universal values z_{il}. We used SBS for this, because it contained 26 of the 33 common variables. Later, we repeated this analysis for IS which contained five of the remaining common variables.

One, relatively simple, approach to find edit rules is to fit a linear or quadratic regression model to each pair of variables $\left(x_{il_1}^{(p)}, x_{il_2}^{(p)}\right)$, with $x_{il_1}^{(p)}$ acting as independent variable and $x_{il_2}^{(p)}$ as dependent variable:

$$x_{il_2}^{(p)} = \alpha + \beta_1 x_{il_1}^{(p)} + \varepsilon_i \qquad \text{(linear)}, \tag{13a}$$

or

$$x_{il_2}^{(p)} = \alpha + \beta_1 x_{il_1}^{(p)} + \beta_2 \left(x_{il_1}^{(p)}\right)^2 + \varepsilon_i \qquad \text{(quadratic)}. \tag{13b}$$

For economic data, a regression model with heteroscedastic disturbances where the variance is proportional to the independent variable often fits better than a model with homoscedastic disturbances. Therefore, we assumed that $E(\varepsilon_i^2) = \sigma^2 \left(\left| x_{il_1}^{(p)} \right| + 1 \right)$ and applied a weighted least squares estimator. (The term "+1" was added to avoid a problem when $x_{il_1}^{(p)} = 0$.)

Next, we restricted attention to those pairs of variables where the explained variance of the model was large (R^2 greater than some threshold). For each of these combinations of variables, the fitted regression model was used to obtain 95% prediction intervals for $x_{il_2}^{(p)}$ given $x_{il_1}^{(p)}$. The upper and lower bounds of these prediction intervals, say $\hat{x}_{il_2}^{(p,\text{upper})}$ and $\hat{x}_{il_2}^{(p,\text{lower})}$, vary as a non-linear function of $x_{il_1}^{(p)}$ which, however, can typically be approximated well by a linear function, by fitting two new linear regression models to these upper and lower bounds. The resulting fitted regression lines provide a natural upper and lower bound on $x_{il_2}^{(p)}$ given $x_{il_1}^{(p)}$, leading to a linear edit rule for z_{il_1} and z_{il_2} of the form:

$$\hat{\alpha}^{(\text{lower})} + \hat{\beta}^{(\text{lower})} z_{il_1} \leq z_{il_2} \leq \hat{\alpha}^{(\text{upper})} + \hat{\beta}^{(\text{upper})} z_{il_1}. \tag{14}$$

Figure 1 illustrates this approach. The solid blue line indicates the original fitted regression line (in this example based on the linear model (13a)). The dashed blue lines indicate the fitted linear regression lines to the upper and lower bounds of the 95% prediction intervals around the original regression line. Black dots represent data points that lie within their prediction interval, red dots lie outside their prediction interval. In this example, about 4% of all points were colored red, which is slightly less than expected. This may be due in part to our linear approximation to the non-linear prediction intervals, but also because the prediction intervals were computed for the same data on which the original regression model was estimated.

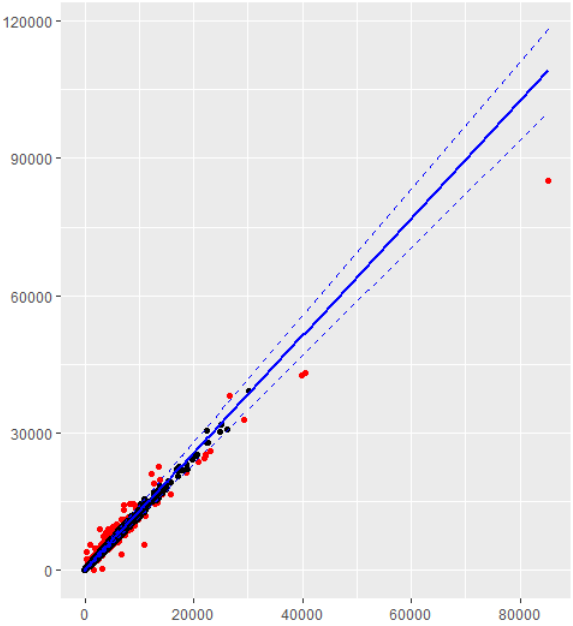


Figure 1. An illustration of the prediction-interval approach.

It should be noted that (14) is a soft edit rule: it is known that some error-free data points will violate this restriction. It may be a step too far to use this edit rule directly to find errors in the observed common variables. However, it seems useful to force any adjustments made during automatic editing to satisfy this restriction, to avoid creating implausible combinations of values. To this end, we introduced edit rules of the form (14) into the error localization problem as soft edit rules by the technique from Scholtus (2015), described near the end of Section 2.1. The weights s_m were assigned to each unit depending on its original observed data. If at least one pair of original observed values $x_{il_1}^{(p)}$ $(p \in B_{il_1})$ or $x_{il_2}^{(p)}$ $(p \in B_{il_2})$ related to a soft edit rule differed from each other by more than $\varepsilon_{l_1} = \varepsilon_{l_2} \approx 2 \times 0.05 = 0.10$ times the smallest value in the pair, then we assigned a large weight $s_m^{(\text{high})}$, indicating that the rule should be treated almost like a hard edit rule. Otherwise, we assigned a small weight $s_m^{(\text{low})}$ so that the rule could be easily ignored during error localization. In this study, we used $s_m^{(\text{high})} = 30$ and $s_m^{(\text{low})} = 0.5$.

After some experiments, we created soft edit rules based on the quadratic regression model (13b) for all pairs of variables with $R^2 \geq 0.95$. The regression models were fitted on SBS and IS data of 2021, i.e., the year before the reference year of the data that were edited in this study. The main regression models (13b) were fitted separately to each NACE group. Within a NACE group, separate models were fitted to the prediction intervals for each size class with at least 10 observations. Hence, separate rules of the form (14) were obtained for each combination of NACE group and size class. In total, 393 different (NACE- and size class-specific) soft edit rules of the form (14) were created: 274 based on SBS data and 119 based on IS data. The average number of soft edit rules applying to a (NACE, size class) pair is 8.02.

In future research, more advanced modeling techniques could also be investigated for deriving soft edit rules. For instance, decision tree models would naturally lead to restrictions of the form (7). In general, machine learning techniques could be useful for finding interesting new edit rules (Dumpert, 2020). Administrative data such as PD may be more useful as input for machine learning than survey data because they contain many more observations, although it should be investigated whether selection bias is an issue.

2.5.2 Deductive correction rules

In total, 14 deductive correction rules for common and source specific variables and 4 deductive rules for reliability weights were derived. We discuss some correction rules in detail.

Margin of turnover, instead of turnover

We define 'margin' as the difference between 'turnover' and 'purchasing value'. After comparing these variables for SBS, STSA and PD analysts concluded that in some cases 'margin' is reported for SBS instead of 'turnover'. The correction rule below is applied in case 'turnover' and 'purchasing value' for SBS compared to STSA and PD is at least 10% smaller and at least 100 000 euro smaller.

```
IF (SBS-margin ≈ PD-margin) AND (STSA-turnover ≈ PD-turnover) AND (SBS-turnover << PD-turnover) AND (SBS-purchasing value << PD-purchasing value) THEN
SBS-turnover = PD-turnover
SBS-purchasing value = PD-purchasing value
```

Note that it is written in pseudo code. We convert each rule in pseudo code to a rule in R-code that is implemented by means of R-package. Furthermore, related variables within SBS (such as total operating income and total operating expenses) are corrected as well to make SBS-data consistent with corrected common variables.

Unjustified wages

A mistake that is made regularly is that a person with sole proprietorship considers its income as wages. However, it should be considered as part of the operating surplus. If there are no SES-data for this enterprise than it is concluded that the enterprise has no employees and SBS-wages should be equal to zero.

IF (legal form = sole proprietorship) AND (SBS-wages > 0) AND (SES-wages is empty) THEN
SBS-operating profit = SBS-operating profit + SBS-wages
SBS-EBITDA = SBS-EBITDA + SBS-wages
SBS-labor costs = SBS-labor costs - SBS-wages
SBS-wages = 0
SBS-employees (in fte) = 0

Again, related variables within SBS are corrected as well.

2.5.3 Reliability weights of common variables

2.5.3.1 Introduction

For reliability weights, we faced a similar issue as for edit rules (see Section 2.5.1): much already tends to be known about the relative reliability of source-specific variables $y_{ij}^{(p)}$ within each individual data source p, but less is known about the relative reliability of the common variables $x_{il}^{(p)}$ and $x_{il}^{(q)}$ as observed in different data sources. At the beginning of the pilot study, subject-matter experts created an initial set of fixed reliability weights $w_l^{(p)}$ for the common variables. Most of these weights were within the range [1, 10], with the exception of the two variables from SES which were considered very reliable and given a weight of 100. However, these fixed weights are considered to be a highly simplified summary of the quality of each common variable. In reality, we expect the reliability of these variables to vary across different subpopulations of units. A single set of weights cannot account for this. However, concrete information about this variation in reliability was lacking.

As noted in Section 2.1, one way to obtain more informative reliability weights would be to take a (large) data set in which the errors are known and model the error probabilities p_j as a function of background variables. In the absence of such a data set, an alternative approach could be to assume a distribution for the true values of each variable and estimate the probability that a particular observed value does not come from this distribution. Given the lack of gold standard edited data for common variables (other than the very small sample from Section 2.4), modelling the error probabilities did not seem like a viable approach in this study. Instead, we focused on three alternative ways to derive unit-specific reliability weights:

- rule-based updating of reliability weights based on patterns found in the manually edited sample from Section 2.4;
- rule-based updating of reliability weights based on indicators of low data quality in individual sources;
- dynamic reliability weights based on a nearest-neighbour method.

These approaches will be discussed further in Sections 2.5.3.2, 2.5.3.3 and 2.5.3.4, respectively.

For all approaches, a relevant question is how much the initial reliability weights $w_l^{(p)}$ defined by the subject-matter experts should be adjusted based on unit-specific information. For a different version of the error localization problem, Freund and Hartley (1967) noted that the absolute values of the weights are not that important; the relative values are more relevant. These authors used weight reduction factors (1/5, 1/10, etc.) to adjust initial weights. We mostly followed this approach, reducing a reliability weight by a certain fixed factor whenever a criterion is satisfied that indicates lower reliability. An alternative approach could have been to define a limited set of possible values for reliability weights and shift a weight to a value with a lower rank for each criterion that is satisfied.

2.5.3.2 Changing reliability weights after confronting data sources

We derived four deductive rules for reliability weights. These are based on findings of analysts that solved inconsistencies for common variables in the sample for NACE 412, 461, 561 and 620. We describe one deductive rule below.

For small companies SBS-turnover is considered less reliable if it differs considerably from other data sources that contain turnover. The following rule is applied.

IF (company size = small) & (STSA-turnover ≈ PD-turnover) & (SBS-turnover <> STSA-turnover) THEN reliability weight of SBS-turnover = 3

Normally, the reliability weight of SBS-turnover equals 5. When it equals 3 than it is lower than the reliability weight of STSA-turnover, which is 4 This may increase the number of records where SBS-turnover is adjusted during automatic editing.

2.5.3.3 Completion quality of SBS-survey

A strategy to refine initial reliability weights for common SBS-variables into a reliability weight per common variable and enterprise is to assess the completion quality of an SBS questionnaire from a respondent. If the completion quality is low for an enterprise then reliability weights of common SBS variables are reduced for this enterprise.

We need indicators of the completion quality of an SBS-respondent. A group of analysts created a list of criteria that can be used:

A) Grouping all other expenses into a single item (housing, energy, advertising, etc.);
B) Other miscellaneous revenue exceeding 50% of total revenue;
C) Total other revenue exceeding 50% of total revenue;
D) Other miscellaneous expenses exceeding 50% of total other expenses;
E) Number of employed persons not in proportion to the size class. Small deviations (one to two size classes) can occur, but more is suspicious;
F) Implausible payrolls, or implausible payroll in proportion to employed persons. Pay particular attention to extreme values, both high and low;
G) Entering the same value for each item, which can include zero values;
H) Entering escalating values for consecutive items.

Criterion F) is not explicitly included, but we take into account all violated internal soft edit rules of SBS. Regarding G) and H), the expectation is that a relatively large number of hard and soft edit rules will be violated, as calculations do not add up and indicators are not plausible. We propose the following criteria.

I) Number of empty items relative to the total number of items for the questionnaire;
J) Number of violated hard edit rules for automatic editing relative to the number of hard edit rules for automatic editing of the specific questionnaire;
K) Number of violated soft edit rules for manual editing relative to the number of soft edit rules for manual editing of the specific questionnaire.

The number of fields and the number of validation rules vary per questionnaire. For example, restaurants have two questionnaires: one for size class 1-3 (with 79 items) and one for size class

4-9 (with 96 items). For the current SBS-process there are 96 hard rules and 68 soft rules for size class 1-3. There are 114 hard rules and 81 soft rules for size class 4-9. The number of violated validation rules in the raw data is generally limited. For size class 1-3, there are a maximum of 9 violated hard rules and 11 violated soft rules. For size class 4-7, there are a maximum of 10 violated hard rules and 12 violated soft rules. Enterprises in size class 8-9 are outside the scope of the study.

For the implementation of criteria I, J and K we derive an edit rule per criterion. When the edit rule for a criterion is violated for an enterprise this means that the criterion applies for that enterprise. The edit rules for criteria I, J and K are

I) *IF (Size class 1-3)*
THEN Number of empty SBS-values ≤ 0.6 × Number of SBS-variables
ELSE Number of empty SBS-values ≤ 0.58 × Number of SBS-variables

J) *IF (Size class 1-3)*
THEN Number of failed edits for automatic editing ≤ 0.06 × Number of edits for automatic editing
ELSE Number of failed edits for automatic editing ≤ 0.05 × Number of edits for automatic editing

K) *IF (Size class 1-3)*
THEN Number of failed soft edits for manual editing ≤ 0.11 × Number of soft edits for manual editing
ELSE Number of failed soft edits for manual editing ≤ 0.085 × Number of soft edits for manual editing

The parameters in the edit rules of format X ≤ *aY* above are determined by means of distributional aspects of X/Y, mainly the tail of the distribution.

For most common variables, the reliability weights (after applying the correction rules in paragraph 2.5.3.2) $w_l^{(p)*}$ are relatively high for SBS. If the completion quality for SBS is lower for enterprise *i* than desired, we reduce some reliability weights for enterprise *i* and SBS. For each criterion, a validation rule is established. Suppose there are *C* rules for determining the completion quality of SBS. These rules indicate whether the completion quality is good. For each rule *c*, the indicator shows whether the rule is violated for enterprise *i* (0 = no; 1 = yes). The indicator also shows whether rule *c* is relevant for common variable *l* (0 = no; 1 = yes). For each criterion, we determine a reduction factor (parameter) for relevant SBS reliability weights. Based on this, we determine:

$$D_{cil} = I_{ci} \times J_{cl} \times V_c$$

$$T_{cil} = D_{cil}, \text{ if } D_{cil} > 0$$
$$T_{cil} = 1, \text{ otherwise.} \qquad (15)$$

$$w_{il}^{(SBS)} = w_l^{(SBS)*} \prod_{c \in C} T_{cil}.$$

Example
For the common variable ‘Net turnover minus excise duties’, the initial reliability weight is 5 for SBS, 4 for STSA, 3 for STSS, and 1.5 for PD. These weights are based on a subjective assessment by a group of analysts and researchers. In the following, we simply refer to ‘turnover’. Suppose

that for an enterprise both STSA-turnover and SBS-turnover are available, but PD-turnover and STSS-turnover are unavailable.

In Table 1, indicators and parameters that apply to an enterprise and SBS-turnover are shown. The reliability weight for this enterprise and SBS-turnover is calculated as: 5 × 1 × 1 × 1 × 1 × 1 × 0.9 × 1 × 0.9 = 4.05. For automatic editing across sources for this enterprise SBS-turnover is still slightly more reliable than STSA-turnover.

Table 1. Criteria for decreasing reliability weights for SBS-variables per enterprise

Criterion	I_{ci}	J_{cl}	V_c	D_{cil}	T_{cil}
1 (A)	1	0	0.9	0	1
2 (B)	0	1	0.9	0	1
3 (C)	0	1	0.8	0	1
4 (D)	1	0	0.9	0	1
5 (E)	0	0	0.9	0	1
6 (I)	1	1	0.9	0.9	0.9
7 (J)	0	1	0.8	0	1
8 (K)	1	1	0.9	0.9	0.9

2.5.3.4 *Dynamic reliability weights based on a nearest-neighbour approach*

Finally, we developed a data-driven way to construct dynamic reliability weights for $x_{il}^{(p)}$. The basic idea of this approach is that a variable for unit i is considered less reliable depending on how much it differs from the same variable for another unit that does not require editing and overall has values that are as similar as possible to unit i. A similar approach to construct dynamic reliability weights was tested previously in a single-source editing context for SBS at Statistics Netherlands, where it worked reasonably well (Scholtus, 2010).

In detail, the proposed method consists of the following steps:

1) Split the data set containing $\left(\mathbf{x}_i^{(1)}, \dots, \mathbf{x}_i^{(P)}\right)$ into, first, records that satisfy all edit rules (11*), (11**) and (7) and, second, records that violate at least one such edit rule. The first set $\mathcal{D}$ – which does not require editing during step 3 – is used as reference data for the second set $\mathcal{R}$.
2) Let $\mathcal{B}_i = \bigcup_l B_{il}$ denote the subset of data sources in which unit i is observed. For each unit $i \in \mathcal{R}$, construct a set of potential donors $\mathcal{D}_i \subseteq \mathcal{D}$ that occur in at least the same subset of data sources: $\mathcal{D}_i = \{d \in \mathcal{D} | \mathcal{B}_i \subseteq \mathcal{B}_d\}$. If possible, we restrict $\mathcal{D}_i$ to contain only donors from the same stratum (NACE group × size class) as unit i. We require $\mathcal{D}_i$ to contain at least M_d donors; if necessary, strata are merged to obtain a sufficient number of donors.
3) For each $i \in \mathcal{R}$, find its nearest neighbour $d_i \in \mathcal{D}_i$ according to the distance function $f(i,d) = \sum_{p \in \mathcal{B}_i} \sum_l \left| \breve{x}_{il}^{(p)} - \breve{x}_{dl}^{(p)} \right|$, where $\breve{x}_{il}^{(p)}$ and $\breve{x}_{dl}^{(p)}$ are standardized versions of $x_{il}^{(p)}$ and $x_{dl}^{(p)}$ so all common variables are a priori equally important for the distance function. We propose to use a robust form of standardization based on the observed median and a robust measure of the spread of values $x_{dl}^{(p)}$ among units in $\mathcal{D}_i$:
   $\breve{x}_{il}^{(p)} = \frac{x_{il}^{(p)} - m_l^{(p)}}{s_l^{(p)}}$;
   $m_l^{(p)} = q_{50\%}\left(x_{dl}^{(p)}; d \in \mathcal{D}_i\right)$;

$s_l^{(p)} = q_U\left(x_{dl}^{(p)}; d \in \mathcal{D}_i\right) - q_L\left(x_{dl}^{(p)}; d \in \mathcal{D}_i\right).$

Choosing $L = 25\%$ and $U = 75\%$ would make $s_l^{(p)}$ the interquartile range. In this study we found that $L = 10\%$ and $U = 90\%$ gave slightly better results, because many common variables contained a large number of zeros, making the interquartile range zero as well. Only common variables with $s_l^{(p)} > 0$ contribute to the above distance function $f(i,d)$; note that $\breve{x}_{il}^{(p)}$ can be computed only for these variables. Any other common variables are treated separately in the next step.

4) For all l that contribute to the minimal distance $f(i,d_i)$, let $r_{il}^{(p)} = \left|\breve{x}_{il}^{(p)} - \breve{x}_{d_i l}^{(p)}\right| / f(i,d_i)$ so that $\sum_{p \in \mathcal{B}_i} \sum_l r_{il}^{(p)} = 1$ and all $r_{il}^{(p)} \in [0,1]$. Denote the number of contributing variables by S_i. For any other common variables that do not contribute, define $r_{il}^{(p)} = 0$ if $x_{il}^{(p)} = m_l^{(p)}$ and define $r_{il}^{(p)} = 1$ otherwise.
5) Update each current reliability weight $w_{il}^{(p)}$ relevant for unit $i \in \mathcal{R}$ by multiplying it by the following factor:

$$\frac{1}{1 + \left\lfloor S_i r_{il}^{(p)} \right\rfloor},$$

where $\lfloor x \rfloor$ equals the largest integer less than or equal to x. Note that, since $f(i,d_i)$ contains S_i terms, the average relative contribution $r_{il}^{(p)}$ equals $1/S_i$. Hence, $S_i r_{il}^{(p)} = r_{il}^{(p)}/(1/S_i)$ indicates the ratio of $r_{il}^{(p)}$ to its average value. Thus, if a variable has a below-average contribution to $f(i,d_i)$, the above factor equals 1 and its reliability weight will not be adjusted. If the relative contribution of a variable lies between a and $a+1$ times the average contribution (where a is a non-negative integer), then its reliability weight is divided by $a+1$.

In this study, we applied this approach with the minimal number of donors set to $M_d = 20$. Donors were selected from both the current year (2022) and the previous year (2021). Still, for 445 units that require multisource editing out of the 9560 units in the pilot study data, it was not possible to find a sufficient number of potential donors with $\mathcal{B}_i \subseteq \mathcal{B}_d$. These are units that occur in many different sources and/or rare combinations, such as SBS and SFLE. For these units, the reliability weights were not updated by the nearest-neighbour approach. In the future, as (top-down) multisource editing becomes a part of regular statistical production, the number of available potential donors from historical data should increase and the number of units for which the nearest-neighbour approach is not feasible should therefore decrease.

2.5.3.5 Other approaches

In addition to the above-mentioned approaches to derive improved reliability weights, we also did some other analyses. The idea was to search for subgroups of units based on background variables for which large inconsistencies with either $x_{il}^{(p)} \gg x_{il}^{(q)}$ or $x_{il}^{(p)} \ll x_{il}^{(q)}$ occur relatively often. To explore these large discrepancies between sources, both interpretable machine learning models and unsupervised clustering techniques were applied. Using features such as NACE code, size class, legal form, complexity of a unit, age of a business, etc., we built decision trees to identify subgroups with large differences between the same variable in two different sources. As an example, one finding was that large differences between SBS and SES on salaries occurred relatively often for units that entered the General Business Register during the reference year, compared to older units. The results of these analyses were discussed with subject-matter experts. If the results could be interpreted to decide in general which of the observed values was more likely to be correct for (some of) these subgroups, then this could be used to adjust the associated reliability weights per subgroup, analogously to Section 2.5.3.2. Unfortunately, although the sessions with subject-matter experts did provide some new insights, it was not possible to derive general rules of this type. That is to say, a general decision which source is more reliable could not be based on background features alone. A separate report of these analyses is provided in Aghaddar and Scholtus (2025).

2.5.4 Imputation methods

In step 4, the universal values z_{il} of the common variables are imputed. A relatively simple procedure was proposed for this in Scholtus et al. (2022). However, in practice not all situations can be handled by this simple procedure. In the current study, the imputation procedure was therefore extended.

We start by deriving, for each unit, lower and upper bounds on each z_{il}, given that restrictions (11*), (11**) and (7) have to be satisfied by the universal values and all values $x_{il}^{(p)}$ that were not set to missing for being erroneous during step 3. The lower bound L_{il} and upper bound U_{il} can be found by solving two linear optimization problems for each l; see Van der Loo and De Jonge (2018, Chapter 8). In our implementation in R, we used the function `detect_boundary_num` from the `validatetools` package to derive these bounds. In some cases, i.e., if the remaining observed values combined with the edit rules provide only limited information or no information at all about z_{il}, it can occur that one or both of L_{il} and U_{il} are unbounded.

Next, each variable z_{il} is imputed by the first method from the list below that is applicable:

1. If at least one of the remaining observed values $x_{il}^{(p)}$ satisfies $L_{il} \leq x_{il}^{(p)} \leq U_{il}$, then impute this value. If multiple feasible observed values are available, use the variable with the largest reliability weight $w_{il}^{(p)}$.
2. If both L_{il} and U_{il} are finite, then impute the midpoint of the feasible interval, i.e. $(L_{il} + U_{il})/2$. (Here, for practical purposes 'finite' is defined as being within the interval $[-M, M]$, with $M = 10^7$ as defined in Section 2.1.)
3. If the universal value of 'net turnover minus excise duties' for unit i has already been imputed by either method 1 or 2, then use a stratum ratio imputation for the remaining z_{il} with this turnover variable as predictor.
4. If the universal value of 'average number of employees in fte' for unit i has already been imputed by either method 1, 2 or 3, then use a stratum ratio imputation for the remaining z_{il} with this number of employees as predictor.
5. Impute the stratum mean.

As in Section 2.5.3.4, stratum was defined in this study as NACE group × size class and units in the set $\mathcal{D}$ (i.e., units that did not require any multisource editing) were used to estimate ratios and means. In the above list, the imputation methods are sorted roughly in decreasing order of the amount of unit-specific information they use. Note that the above approach could be refined in the future. In particular, for some common variables (e.g., wages) it may be better to switch the order of methods 3 and 4.

The imputed values for $\mathbf{z}_i$ obtained by this approach do not necessarily satisfy all edit rules (11*), (11**) and (7). Therefore, they were minimally adjusted if necessary. (Recall that the Fellegi-Holt based error localization used in step 3 guarantees that a set of values always exists for $\mathbf{z}_i$ that satisfies all edit rules, given the remaining values in $\mathbf{x}_i^{(1)}, \dots, \mathbf{x}_i^{(P)}$.) For the adjustment procedure, we used a linear minimization problem, which was implemented by re-using functionality from the `errorlocate` package.

2.5.5 Scenarios

We implemented several methods and correction rules for automatic editing. To assess the individual effect of these methods and rules on the final results we distinguish five different scenarios.

1. All implemented methods are used. Output for evaluation criteria includes a separate effect for use of correction rules for variables, so we can distinguish
    a. Only correction rules for variables (i.e., the status after step 1)
    b. All methods (i.e., the status after step 5)
2. Without use of correction rules (Section 2.5.2)
3. Without use of dynamic weights (Section 2.5.3.4)
4. Without use of dynamic weights (Section 2.5.3.4) and without adjusted reliability weights because of low completion quality of SBS-survey (Section 2.5.3.3)
5. Without use of soft edit rules (Section 2.5.1)

By means of these scenarios we can assess the following effects:

- Correction rules: Compare scenario 2 with scenario 1b
- Dynamic weights: Compare scenario 3 with scenario 1b
- Adjusted reliability weights because of low completion quality of SBS-survey: Compare scenario 4 with scenario 3
- Soft edit rules: Compare scenario 5 with scenario 1b

2.6 Evaluating the quality of automatic multisource editing

We would like to evaluate how well the automatic data editing is performing. We distinguish between two ways of evaluation. The first way is to draw a sample of units and ask analysts to carefully edit those records. We can then compare the automatically edited data with the manually edited data. The second way is to use evaluation measures without the use of an audit sample. Both are further described below.

2.6.1 Evaluation using the sample

The sample selection and the difficulties encountered during manual editing were discussed in Section 2.4. For the four economic sectors where a sufficiently large subsample of ‘gold standard’ data was available, the quality of automatic error localization can be evaluated by comparing the error patterns found by automatic editing to the error patterns found by manual editing. Evaluation measures based on the number of false positives (incorrect errors) and false negatives (missed errors) can be computed, such as recall, precision, and the F1-score. The latter is the weighted harmonic mean of precision and recall. Recall is the fraction of true errors found during error localization; precision is the fraction of errors found during error localization that are true errors (not false positives).

In general, it would also be useful to compare the distributions of values after automatic and manual editing, by measures such as the average absolute distance and the absolute or relative difference in means. These measures reflect the combined quality of error localization and imputation. EDIMBUS (2007, Appendix D) provides a large set of evaluation measures for (automatic) editing; see also De Waal et al. (2011, Chapter 11). However, it was decided in this study that these distribution-based measures could not be estimated in a meaningful way, given that only 69 ‘gold standard’ edited cases were available in total across the four economic sectors.

Table 2. Evaluation results of error localization for common variables among all sources.

NACE 2008	scenario	TP	FN	FP	TN	recall	precision	F1
412	1a	6	12	0	194	0.333	1.000	0.500
	1b	9	9	7	187	0.500	0.562	0.529
	2	4	14	5	189	0.222	0.444	0.296
	3	8	10	6	188	0.444	0.571	0.500
	4	8	10	6	188	0.444	0.571	0.500
	5	9	9	5	189	0.500	0.643	0.562
461	1a	1	16	7	280	0.059	0.125	0.080
	1b	3	14	8	279	0.176	0.273	0.214
	2	2	15	0	287	0.118	1.000	0.211
	3	2	15	8	279	0.118	0.200	0.148
	4	2	15	8	279	0.118	0.200	0.148
	5	3	14	8	279	0.176	0.273	0.214
561	1a	18	37	3	577	0.327	0.857	0.474
	1b	26	29	10	570	0.473	0.722	0.571
	2	17	38	5	575	0.309	0.773	0.442
	3	23	32	13	567	0.418	0.639	0.505
	4	21	34	12	568	0.382	0.636	0.477
	5	26	29	10	570	0.473	0.722	0.571
620	1a	19	78	2	724	0.196	0.905	0.322
	1b	27	70	4	722	0.278	0.871	0.422
	2	14	83	1	725	0.144	0.933	0.250
	3	27	70	4	722	0.278	0.871	0.422
	4	27	70	4	722	0.278	0.871	0.422
	5	27	70	4	722	0.278	0.871	0.422
total	1a	44	143	12	1775	0.235	0.786	0.362
	1b	65	122	29	1758	0.348	0.691	0.463
	2	37	150	11	1776	0.198	0.771	0.315
	3	60	127	31	1756	0.321	0.659	0.432
	4	58	129	30	1757	0.310	0.659	0.422
	5	65	122	27	1760	0.348	0.707	0.466

In Table 2 and Table 3 we show evaluation criteria on error localization for all common variables and only for common variables that are observed in SBS, respectively. For NACE (2008) 412 recall and precision is lower for common SBS-variables than for all common variables. A possible reason is that the analysts for NACE 412 made some divergent choices for correction, while we applied the same correction rules for all NACE. However, if we do not apply the correction rules then recall and precision is even lower for common SBS-variables.

For NACE 461 recall is better for common SBS-variables than for all common variables. However, precision is worse for scenario 1b, 3, 4, and 5 if we make that comparison. For NACE 561 recall and precision improve somewhat if we only consider common SBS-variables and only use correction rules. For NACE 620 recall and precision are worse for common SBS-variables than for all common variables, for almost all scenarios. For all considered NACE combined this is the case for all scenarios.

Table 3. Evaluation results of error localization for common SBS-variables.

NACE 2008	scenario	TP	FN	FP	TN	recall	precision	F1
412	1a	2	5	0	133	0.286	1.000	0.444
	1b	3	4	5	128	0.429	0.375	0.400
	2	0	7	3	130	0.000	0.000	0.000
	3	2	5	4	129	0.286	0.333	0.308
	4	2	5	4	129	0.286	0.333	0.308
	5	3	4	3	130	0.429	0.500	0.462
461	1a	1	7	7	205	0.125	0.125	0.125
	1b	2	6	8	204	0.250	0.200	0.222
	2	1	7	0	212	0.125	1.000	0.222
	3	1	7	8	204	0.125	0.111	0.118
	4	1	7	8	204	0.125	0.111	0.118
	5	2	6	8	204	0.250	0.200	0.222
561	1a	16	28	2	394	0.364	0.889	0.516
	1b	21	23	9	387	0.477	0.700	0.568
	2	12	32	5	391	0.273	0.706	0.393
	3	18	26	9	387	0.409	0.667	0.507
	4	16	28	6	390	0.364	0.727	0.485
	5	21	23	9	387	0.477	0.700	0.568
620	1a	10	59	2	509	0.145	0.833	0.247
	1b	13	56	3	508	0.188	0.812	0.306
	2	3	66	0	511	0.043	1.000	0.083
	3	13	56	3	508	0.188	0.812	0.306
	4	13	56	3	508	0.188	0.812	0.306
	5	13	56	3	508	0.188	0.812	0.306
total	1a	29	99	11	1241	0.227	0.725	0.345
	1b	39	89	25	1227	0.305	0.609	0.406
	2	16	112	8	1244	0.125	0.667	0.211
	3	34	94	24	1228	0.266	0.586	0.366
	4	32	96	21	1231	0.250	0.604	0.354
	5	39	89	23	1229	0.305	0.629	0.411

For each scenario recall is relatively high for NACE 561, both for all common variables and all common SBS-variables. For almost all scenarios (except for 1a) precision is relatively high for NACE 620. The use of soft edit rules does not improve recall or precision. For NACE 421 the precision is even lower when soft edit rules are used.

Overall, the scenarios with the highest recall are scenario 1b and 5, and the scenario with the highest precision is scenario 1a. The highest F1-score is obtained for scenario 5, while it should be noted that the F1-score for scenario 1b is only marginally lower. The ‘total recall’ for scenario 1b (use all methods) is 0.305 for common SBS-variables and 0.348 for all common variables. There are several reasons why ‘total recall’ is not higher.

1. Automatic editing by means of the Fellegi-Holt paradigm means that we minimize the sum of reliability weights of variables for which a value is adjusted. This often implies that a minimal number of values is adjusted. With manual editing more values are generally adjusted, especially in the case of systematic errors.

2. Editors have access to information that is not in our data set, such as data from annual accounts, information about monthly changes in the structure of an enterprise and NACE values of underlying legal units.
3. Each editor may have a different editing strategy, while we use general correction rules.

For scenario 2, the number of false negatives is relatively high and the number of false positives is relatively low. That is, including correction rules in our automatic editing process

a) decreases the number of false negatives
b) increases the number of false positives.

Ad a) Part of the correction rules resolve systematic errors, which some analysts have also resolved. These are errors that cannot (simply) be resolved using the Fellegi-Holt paradigm, because it implies that several values have to be adjusted.

Ad b) Many corrections made by analysts have not been coordinated. That is, one analyst may correct certain errors in a well-argued way, but others may not. Since our correction rules are applied for all NACE they generate false positives. Records with obvious incorrect manual corrections were removed from the sample, but there was only a retrospective look at errors that should have been manually corrected.

Recall improves for NACE 561 (restaurants) by adjusting reliability weights for low completion quality of SBS-survey. However, precision for NACE 561 decreases for common SBS-variables in that case. The use of dynamic weights improves recall and precision considerably for NACE 412, 461, and 561.

To put the above recall and precision values in context, it is interesting to compare them to previous evaluations of the quality of the current production system for automatic editing of SBS data at Statistics Netherlands (no multisource editing). Bergsma (2022) evaluated SBS 2019 data and found a recall of 0.118 and a precision of 0.551 (F1-score = 0.194; based on a sample of 932 units). The current results for multisource editing achieved both a better recall and a better precision. Earlier, Scholtus and Göksen (2012) reported a recall of 0.449 on a small subset of 10 variables for a sample of 580 units from SBS 2007 on wholesale trade; precision was not reported in that study. It should be noted that these 10 variables constitute the employment block on the SBS questionnaire, for which relatively strong edit rules are available during automatic editing compared to other SBS variables. A limitation of both previous evaluations is that the sampled cases were edited manually as part of regular SBS production. Since selective editing is used for the Dutch SBS, these cases are not necessarily representative of all units that require editing.

2.6.2 Evaluation without a sample

Some other results are reported here as a more indirect way of evaluating the effects of automatic editing. First, one practical issue is the computation time of error localization (steps 3 and 5), which is the most computationally intensive part of the proposed procedure. The following results all refer to scenario 1. The error localization problem for common variables (step 3) was solved successfully for all 9560 units. For 8957 units (93.7%), an optimal solution was found within the time limit of 30 seconds; for the remaining 603 units (6.3%), a solution was found but its optimality could not be established within 30 seconds. The subsequent error localization problem for SBS variables (step 5) was solved to guaranteed optimality within 30 seconds for 9239 units (96.6%) and a possibly sub optimal solution was found for a further 200 units (2.1%); for the remaining 121 units (1.3%), a processing error occurred somewhere during step 4 or 5.

Most of these errors appear to be due to numerical issues and could be resolved by improving the current prototype implementation. The average computation time for error localization per unit was 2.29 seconds in step 3 and 1.02 seconds in step 5.

A second interesting result is that some common variables required much more automatic editing than others. The following variables were edited most often:

- average number of employees in fte (SBS): edited for 24.1% of units;
- net turnover minus excise duties (STSA): edited for 18.8% of units;
- labor costs (PD): edited for 18.7% of units;
- wages (SBS): edited for 18.5% of units;
- labor costs (SBS): edited for 17.1% of units;
- sales margin (PD): edited for 13.7% of units
- purchasing value (PD): edited for 12.0% of units;
- wages (PD): edited for 11.4% of units;
- self-manufactured turnover in the Netherlands minus excise duties and allocated freight costs (SBS): edited for 10.4% of units.

All other common variables were edited for less than 10% of units (most of them less than 5%). We intend to follow up on a sample of cases with often-edited variables with subject-matter experts, to assess whether these edits are plausible. This remains to be done.

To analyse the changes made to the data during automatic editing, it is helpful to visualize the data. As an example, Figure 2 shows the bivariate distribution of the same common variable ('net turnover minus excise duties') in two sources (SBS and STSA) before editing [panel (a)] and after editing [panel (b)], using scenario 1. The color of each point indicates whether a pair of values was changed and, if so, in which source(s). The shape of each point indicates whether changes occurred only during deductive correction (step 1) or also because of error localization (steps 3 and 5). In panel (a), it is seen that the observed data contained a large number of inconsistencies for this variable, i.e., deviations that were large enough both in relative and absolute terms to yield violations of edit rule (11*). It is also clear that the STSA variable was edited more often than the SBS variable. This is not unexpected, as the STSA variable has a lower reliability weight than the SBS variable (4 and 5, respectively) in the initial set of weights specified by subject-matter experts. Nearly all changes occurred during error localization; in fact, only one deductive correction rule has been specified that can affect the SBS variable and no deductive correction rules affect the STSA variable. The data after editing [panel (b)] mostly look as expected. Among large turnover values, only relatively small deviations between the two sources are left. However, edit rule (11*) allows for deviations of up to about €100 000 between a pair of observed values, and these can be seen near the origin of the plots in panel (b), in particular for NACE groups 461 and 620. (Recall that $\log_{10} 100 = 2$.) A few unexpected larger differences remain after automatic editing. In a production-setting, these could be followed-up during top-down manual editing.

As another example, Figure 3 shows the bivariate distribution of derived universal values (z_{il}) for two related common variables: 'net turnover minus excise duties' and 'production'. Green points belong to the set of potential donors $\mathcal{D}$ as defined in Section 2.5.3, i.e., these are units for which hardly any multisource editing is expected to be necessary. Red points refer to all other units, i.e., units with at least one large inconsistency between sources. Panel (a) shows the data obtained under scenario 1 and panel (b) the data under scenario 5, without the additional soft edit rules that were derived in Section 2.5.1.

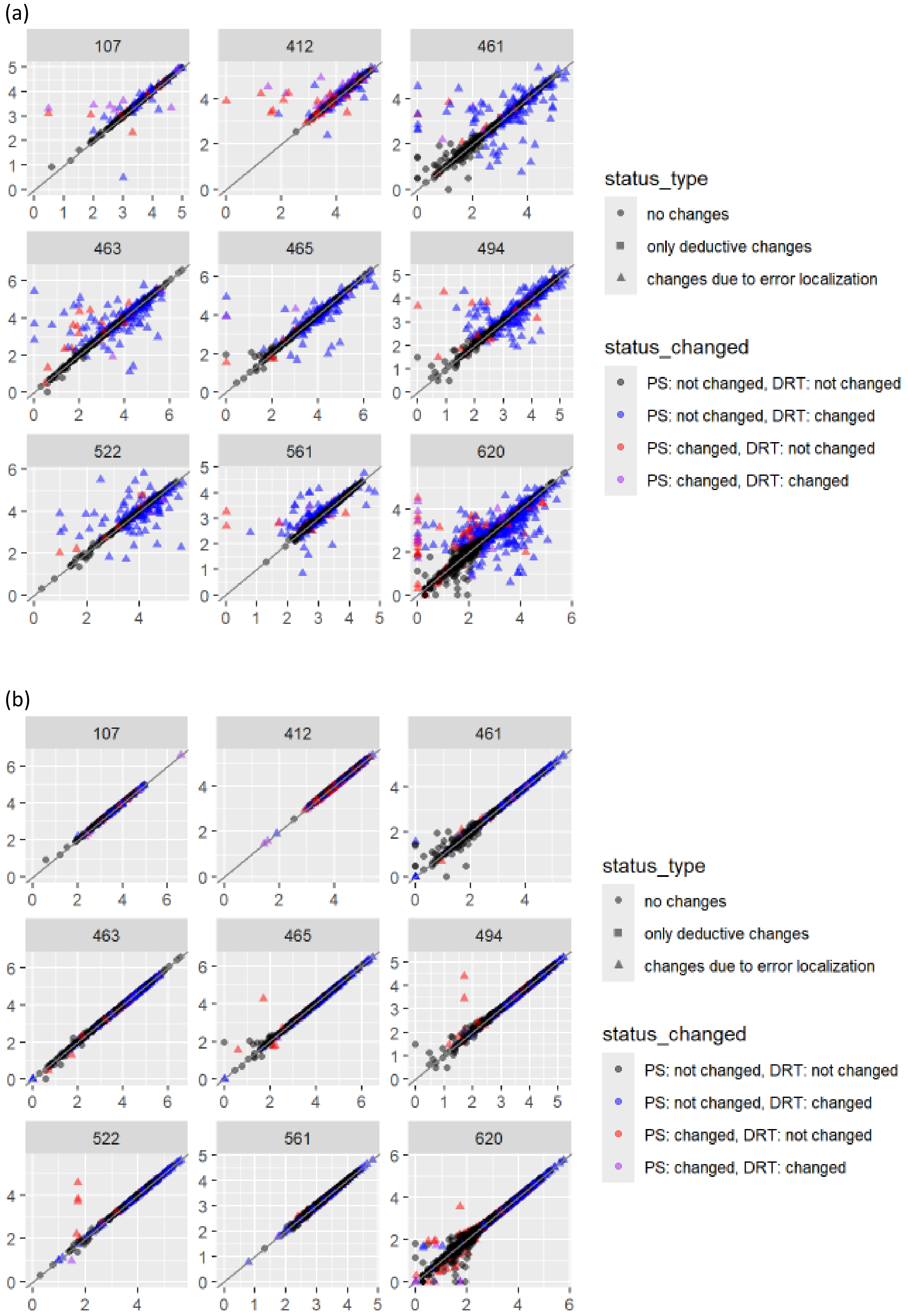


Figure 2. Comparison of common variable ‘net turnover minus excise duties’ in SBS (horizontal axis; Dutch acronym = PS) and STSA (vertical axis; Dutch acronym = DRT). Values are shown as multiples of €1000 on a log 10 scale. Panel (a) shows the data before automatic multisource editing; panel (b) shows the data after automatic multisource editing.

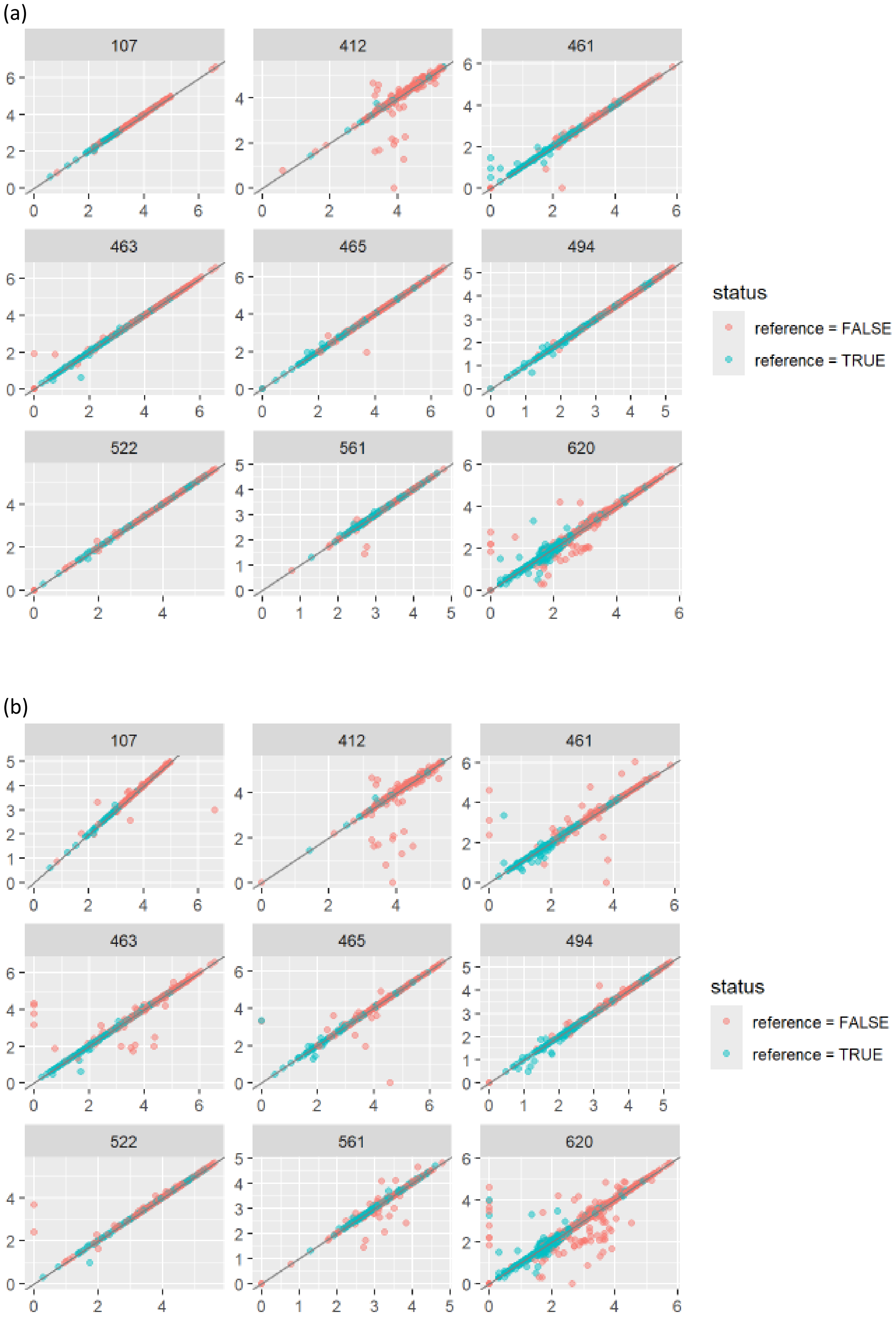


Figure 3. Comparison of universal values of the common variables ‘net turnover minus excise duties’ (horizontal axis) and ‘production’ (vertical axis). Values are shown as multiples of €1000 on a log 10 scale. Panel (a) shows scenario 1; panel (b) shows scenario 5 (no soft edit rules).

It is seen in panel (b) that without the additional soft edit rules in scenario 5, several large differences occurred between the derived universal values of ‘net turnover minus excise duties’ and ‘production’ among the red points (units that required editing). In particular, these differences would be unusually large compared to the ‘naturally occurring’ differences observed among green points. Under scenario 1 [panel (a)], these large differences where much rarer and the distributions of red and green points mostly appear to be more similar. NACE group 412 is an exception; for this group no soft edit rules relating ‘net turnover minus excise duties’ and ‘production’ were created, because the explained variance in the estimated regression models of the form (13b) was too small ($R^2 = 0.94$ or $R^2 = 0.77$, depending on which variable is chosen as target variable). Figure 3 illustrates that the additional soft edit rules in scenario 1 are useful to improve the plausibility of automatically edited data. It is also seen that the criteria used to define these soft edit rules (in particular, the bound on explained variance) could still be improved.

2.7 Refactoring the R code

As part of the EBS Grant, we have refactored and significantly extended the R code that was previously developed in the pilot study of 2021-2022. Improvements to the code include:

- We have added code for steps 1 and 2 of the multisource editing process, which were not considered in the earlier pilot study.
- We have split the code for steps 3, 4 and 5 into separate scripts, so we could more easily test the impact of changes in each of the steps on the outcomes.
- An option for parallel processing has been added for the most time-consuming steps, in particular steps 3, 4 and 5. This allows data of different units to be edited simultaneously.
- In the original code, many rules and parameter settings were hard-coded and only applicable to the specific data set at hand. The new version of the code is much more generic. Edit rules, correction rules, rules for adjusting reliability weights, parameters such as ε_l^* and Δ_l^*, definitions of strata and schemas to merge strata in case of insufficient donors are now all specified outside of the code in separate input files, which makes it much easier to change a rule or parameter, without having to change the rest of the code.
- As a further extension of the previous point, an option has been added to switch certain parts of the multisource editing process ‘on’ or ‘off’, e.g., the different methods from Section 2.5.3 to adjust reliability weights in step 2. This made it possible to run different scenarios as discussed in Section 2.5.5.

One important practical outcome of the above improvements is that the code has become much faster. Using the original code from the study of 2021-2022 it would have taken several days (if not weeks) to apply multisource editing to the 9560 units considered here. (No attempt has been made to actually run the code from 2021-2022 on a data set of this size.) With the new code this took less than ten hours, using six cores for parallel processing.

A public version of the R code for automatic multisource editing developed under this grant is available at https://github.com/SNStatComp/MultiSrcAutoEdit.

2.8 Discussion and conclusion

By means of a sample of data for different sources for 2022 and with the help of a group of analysts, a considerable number of inconsistencies in common variables among sources has been solved. The analysts substantiated the corrections they made. This resulted in a set of deductive correction rules, improved reliability weights for common variables, criteria to assess the completion quality of SBS surveys, an improved setup for automatic multisource editing, and R-code at GitHub where rules and methods are implemented and evaluated.

In Scholtus et al. (2024) and also in the intermediate report for WP1, we described a three-step plan for automatic multisource editing. In the current report we have added two preliminary steps, namely for 1) deductive correction of values for common variables and source-specific variables, and 2) derivation of unit-specific reliability weights. A third preliminary step (step 0) may consist of correction of obvious systematic errors, such as 1000-errors, in individual sources. Another improvement in the setup is that consistency rules do not only check if the relative difference between two sources is too large, but also if the absolute difference is acceptable.

The deployment of analysts has resulted in an increased insight in the reliability of data sources (both surveys and registers). By comparing common variables across data sources we discovered several systematic errors that occur frequently. This can already be useful for the production of SBS 2024. It turns out that it is important that analysts have knowledge about business surveys, business administration, tax forms and tax rules. With sufficient knowledge, part of the inconsistencies for common variables can be solved without contacting respondents.

There are several things that can be improved. Contacting respondents about inconsistencies may result in additional correction rules and further improvement of reliability weights per common variable, data source, and enterprise or NACE / size class. Sessions for analysts about business administration, tax forms and tax rules are important such that more ‘gold standard’ data can be obtained. These data can be used to further improve methods for automatic multisource editing.

In particular, this is relevant for step 5 of the proposed approach (automatic editing within each individual data source). Relatively little information that is relevant for this step was obtained from manual editing in this study and therefore the results of this step could not yet be evaluated in detail. This remains an issue to be improved in the near future. As a next step, we intend to use historical manually edited SBS data to learn more about the way corrections to common variables from steps 3 and 4 should be propagated to source-specific variables in SBS during step 5.

Some parameters that are used in edit rules, for instance for assessment of completion quality of SBS surveys by (15), are not optimal yet. This can be realized by examining a number of filled-out SBS-questionnaires that (almost) meet criteria for low completion quality. Furthermore, it is important to submit results of automatic multisource editing to content-related experts for evaluation purposes.

Finally, a recent development at Statistics Netherlands that is relevant for the output of this study, is the proposed introduction of a generic processing system for business statistics. This system, which is currently being developed, will be used to produce many different business statistics. Each statistical process can specify its own ‘production strategy’ by means of orchestration (which functions to use and in which order), rules and parameter settings for the use of the generic statistical methods that are available in the system. According to current plans, IS and SBS will be the first statistics to be produced using the generic processing system, starting in 2026. For multisource automatic editing at Statistics Netherlands, a next step is therefore to work out how the proposed methodology can be made available within the generic processing system. Once this is done, parts of the method could be used easily (at least from a technical point of view) by many statistical processes. In particular for SBS and IS, we expect that the deductive correction rules for common variables that were derived in this study will soon be implemented via the generic processing system. One issue that remains to be investigated is whether these rules still need fundamental adjustments or expansions to be applicable to other NACE groups.

2.9 References

M. Aghaddar and S. Scholtus (2025), Multisource Automatic Editing: A Data-Driven Approach to Variable Consistency and Rule-Based Correction. Internal report, Statistics Netherlands, The Hague. Available on request.

M. Bankier (2006), Imputing Numeric and Qualitative Variables Simultaneously. Memo, Statistics Canada, Social Survey Methods Division.

F. Bergsma (2022), Verbeteren van de foutlokalisatiemethode voor automatische controle en correctie van bedrijfsstatistieken. Internship report (in Dutch), Statistics Netherlands, The Hague.

B. Chen, Y. Thibaudeau and W.E. Winkler (2003), A Comparison Study of ACS IF-Then-Else, NIM, DISCRETE Edit and Imputation Systems using ACS Data. UNECE Work Session on Statistical Data Editing, Madrid.

J. Daalmans and S. Scholtus (2018), A MIP Approach for a Generalised Data Editing Problem. Discussion Paper, Statistics Netherlands, The Hague, available here.

J.-C. Deville and C.-E. Särndal (1992), Calibration Estimators in Survey Sampling. *Journal of the American Statistical Association* **87**, 376–382.

T. de Waal, J. Pannekoek and S. Scholtus (2011), *Handbook of Statistical Data Editing and Imputation*. John Wiley & Sons, Hoboken, NJ.

F. Dumpert (2020), Theme Report of the Editing & Imputation Group. Report, UNECE HLG-MOS Machine Learning Project.

EDIMBUS (2007), *Recommended Practices for Editing and Imputation in Cross-Sectional Business Surveys*. Eurostat manual prepared by ISTAT, Statistics Netherlands, and SFSO.

I.P. Fellegi and D. Holt (1976), A Systematic Approach to Automatic Edit and Imputation. *Journal of the American Statistical Association* **71**, 17–35.

R.J. Freund and H.O. Hartley (1967), A Procedure for Automatic Data Editing. *Journal of the American Statistical Association* **62**, 341–352.

G.E. Liepins (1980), A Rigorous, Systematic Approach to Automatic Data Editing and its Statistical Basis. Report ORNL/TM-7126, Oak Ridge National Laboratory.

R.J.A. Little and P.J. Smith (1987), Editing and Imputation of Quantitative Survey Data. *Journal of the American Statistical Association* **82**, 58–68.

F. Rocci (2020), Machine Learning for Data Editing Cleaning in NSI (Editing & Imputation): Some Ideas and Hints. Report, UNECE HLG-MOS Machine Learning Project.

S. Scholtus (2010), Betrouwbaarheidsgewichten voor het automatisch gaafmaken bij de Productiestatistieken. Internal report (in Dutch), Statistics Netherlands, The Hague.

S. Scholtus and S. Göksen (2012), Automatic Editing with Hard and Soft Edits – Some First Experiences. UNECE Work Session on Statistical Data Editing, Oslo.

S. Scholtus (2015), New Results on Automatic Editing using Hard and Soft Edit Rules. UNECE Work Session on Statistical Data Editing, Budapest.

S. Scholtus, W. de Jong, A. Vaasen-Otten and F. Aelen (2022), Towards a New Integrated Uniform Production System for Business Statistics at Statistics Netherlands: Automatic Data Editing with Multiple Data Sources. UNECE Expert Meeting on Statistical Data Editing, 3-7 October 2022 (virtual).

S. Scholtus, A. van Delden, R. Willems and F. Aelen (2024). Current work on automatic multisource editing at Statistics Netherlands. UNECE Expert meeting on Statistics Data Editing, 7-9 October, Vienna.

M. van der Loo and E. de Jonge (2018), *Statistical Data Cleaning with Applications in R*. John Wiley & Sons, Hoboken, NJ.

M. van der Loo and E. de Jonge (2021), Data Validation Infrastructure for R. *Journal of Statistical Software* **97** (10), 1–31.

3. Dashboard for top-down analysis

3.1 Introduction

In addition to multisource automatic data editing, some manual data editing will always be necessary. In order to do this most efficiently, in an earlier pilot study at CBS (Vaasen-Otten et al., 2022) score functions were developed to aid top-down analysis by identifying potential influential errors. Similar score functions can also be used for top-down analyses across statistics, to determine potential influential inconsistencies between statistics.

A typical score function for comparing statistics is the difference in value between the two statistics (or between a statistic and administrative data), multiplied by a weight reflecting how often the unit contributes to the compilation of output-level figures – for example, as a donor in imputations or through a sampling weight. The inclusion weight often serves as a good approximation. Finally, the result is divided by an aggregate total to account for the impact of the inconsistency at the output level.

In the pilot study, we implemented these score functions in a consistency dashboard starting from the unit-level scores, which we then aggregated to higher levels to indicate whether there are significant inconsistencies within an (output) aggregate. This way, it is possible to quickly and efficiently drill down from the aggregate levels to the most influential inconsistencies at the unit level. The dashboard was subsequently used by analysts to gain some experience with real production data from multiple statistics. Results of the pilot study were highly encouraging, prompting us to move forward with incorporating it into regular production, also for more statistics than those used in the pilot. However, expansions to the dashboard are needed for that.

The goal of this part of work package 1 is to further enhance the consistency dashboard, making it more flexible and more broadly applicable. The objective is to enable its use with a wide range of relevant business statistics or other data sources, regardless of the underlying unit types such as enterprise groups, enterprises, local units, VAT units, etc. Additionally, the dashboard should support different levels of aggregate output, including NACE sections, divisions, groups, classes, or other relevant output categories. We also want to ensure that the dashboard aligns with the automatic editing process, for instance by visualizing influential changes made by automatic editing within and across statistics. We evaluate the extent to which we achieve these goals by assessing how easy it is to add an additional business statistic or data source to the dashboard, or use a different type of unit or output level.

3.2 Approach to make the dashboard more flexible and broadly applicable

To make the consistency dashboard that was developed in the pilot study more flexible and broadly applicable, we implemented the following approach:

- Make a clear separation between data standardization, data preparation, data processing, and displaying data on a dashboard. Data come from different surveys and administrative sources and are first converted into a standard format. These standardized data are then further prepared for different processing steps, such as calculating score functions on unit and aggregate levels, after which they can be displayed on various screens in a dashboard. Please note that data standardization is not included in the codebase of the dashboard we provide;
- Users – or preferably a data administrator – must pre-fill several tables in the database prior to data preparation and data processing:
    - A standardized *input data table* containing all input values for each statistic or source of administrative data;

    - A *population table* listing all statistical units, along with relevant metadata such as legal names, NACE codes, and associated unit types (e.g., Enterprise Group, Enterprise, Kind-of-Activity Unit);
    - A *generic mapping table* that defines the relationships between the units of different types, indicated as parent company versus subsidiary. Note that currently only 1-to-n relationships are supported;
- We aim to work in a metadata-driven manner as much as possible, which means that data preparation and data processing are controlled and governed by metadata. For this purpose, dedicated auxiliary tables are used to list the following metadata components:
    - *Periods* – e.g., different months, quarters, years;
    - *Statistics* – e.g., statistical surveys such as SBS, Prodcom, STS, or administrative sources like VAT;
    - *Variables* – e.g., all variables that appear in more than one source, including those that may be derived from more detailed variables;
    - *Unit types* – e.g., Enterprise Group (EG), Enterprise (ENT), Kind-of-Activity Unit (KAU), etc. Note that currently only one unit type is supported per statistic. If a statistic relies on multiple unit types (e.g., for different subpopulations), each combination should be treated as a separate statistic;
    - *(Output) Aggregate types* – e.g., industry sectors, regions, or other aggregation groupings used in output statistics;

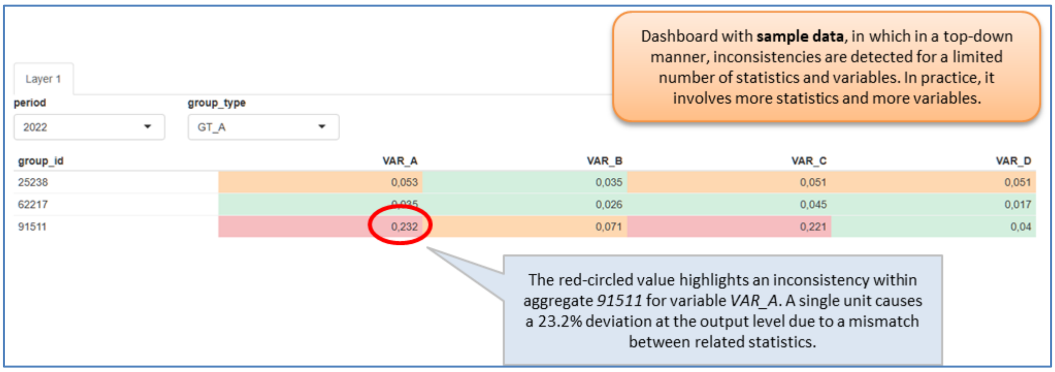
  The user-pre-filled tables must exclusively contain metadata elements as specified in the respective auxiliary tables;
- Using the generic mapping table, data from underlying subsidiary units are aggregated during data preparation to the level of the parent company unit, so that statistics based on different types of units can still be compared with each other. This is done only in case data from all underlying subsidiary units is available (so possible imputations should be done prior to filling the data input table);
- When using multiple data sources based on different unit types, decisions must always be made on how to handle these differences when comparing the sources. Various situations can arise that make the confrontation process considerably more complex than simply calculating scores. Here are some examples:
    - Two statistics based on the same unit type can be compared directly;
    - Comparing a statistic based on, for example, an Enterprise (ENT) with one based on an Enterprise Group (EG) is only valid if the EG is equivalent to the ENT (i.e., the EG consists of just one ENT). Note: this concerns the potential impact of an inconsistency on the first statistic in the comparison;
    - Conversely, when comparing an EG-based statistic with an ENT-based one (i.e., assessing the possible effect of the inconsistency on the EG statistic), the underlying ENTs belonging to the EG must be aggregated (ignoring consolidations);
    - When aggregating ENTs to EG level, care must be taken not to compare those aggregated values with other ENT-level statistics as if they both were on the level of the EG.

  These decisions/logistics have already been incorporated into the data preparation and data processing for the dashboard;
- Calculating the scores themselves is, in terms of the formula, quite simple – we use the most basic version from Vaasen-Otten et al. (2022). However, one important aspect is that the denominator of the score requires a total value for the output aggregate, which differs per statistic. In some cases, one statistic may already have a more robust total available than another, so a choice must be made as to which total to use. All relevant total values are precomputed during the data preparation phase. An additional helper table needs to be filled by the user, specifying, for each variable, which data source should be used in the denominator of the scores;
- We included the option to exclude certain units from the calculation of aggregate scores that are used in the top-down analysis, because they are managed by other analysts (e.g. by the Large Cases Unit);

- The dashboard has been intentionally kept as simple as possible: all data shown is precomputed during the data processing phase. The dashboard's sole purpose is to present this information in a clear and structured way, without performing any computations itself.

3.3 The consistency dashboard

The consistency dashboard follows a top-down structure and is organized into four layers. The screenshots below illustrate these four layers using an example dataset (included in the GitHub repository along with the code). For clarity, the example dataset has been kept relatively small, with a limited number of periods, statistics, variables, units, and output aggregates, which are all fictitious. Additional explanations are provided within the screenshots using text balloons.



period: 2022

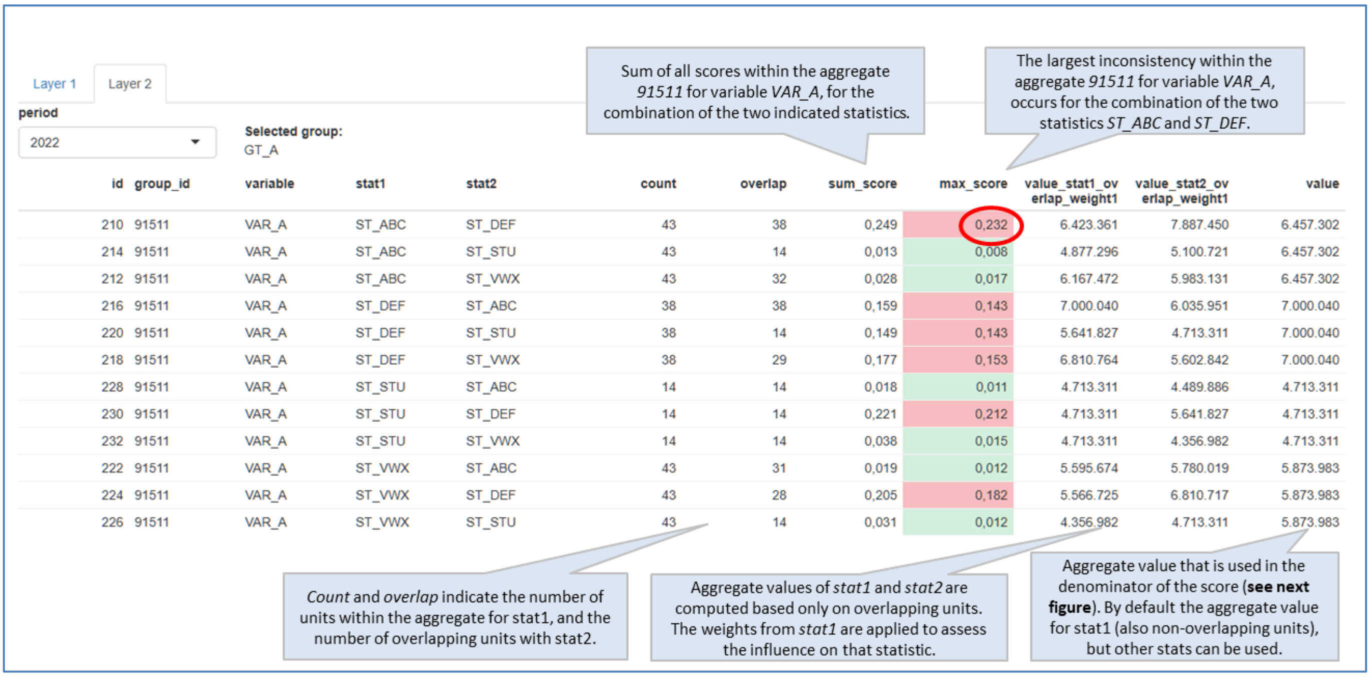
group_type: GT_A

group_id	VAR_A	VAR_B	VAR_C	VAR_D
25238	0,053	0,035	0,051	0,051
62217	0,035	0,026	0,045	0,017
91511	0,232	0,071	0,221	0,04

Dashboard with **sample data**, in which in a top-down manner, inconsistencies are detected for a limited number of statistics and variables. In practice, it involves more statistics and more variables.

The red-circled value highlights an inconsistency within aggregate *91511* for variable *VAR_A*. A single unit causes a 23.2% deviation at the output level due to a mismatch between related statistics.

Figure 4. Dashboard: overview.



period: 2022

Selected group:
GT_A

id	group_id	variable	stat1	stat2	count	overlap	sum_score	max_score	value_stat1_overlap_weight1	value_stat2_overlap_weight1	value
210	91511	VAR_A	ST_ABC	ST_DEF	43	38	0,249	0,232	6.423.361	7.887.450	6.457.302
214	91511	VAR_A	ST_ABC	ST_STU	43	14	0,013	0,008	4.877.296	5.100.721	6.457.302
212	91511	VAR_A	ST_ABC	ST_VWX	43	32	0,028	0,017	6.167.472	5.983.131	6.457.302
216	91511	VAR_A	ST_DEF	ST_ABC	38	38	0,159	0,143	7.000.040	6.035.951	7.000.040
220	91511	VAR_A	ST_DEF	ST_STU	38	14	0,149	0,143	5.641.827	4.713.311	7.000.040
218	91511	VAR_A	ST_DEF	ST_VWX	38	29	0,177	0,153	6.810.764	5.602.842	7.000.040
228	91511	VAR_A	ST_STU	ST_ABC	14	14	0,018	0,011	4.713.311	4.489.886	4.713.311
230	91511	VAR_A	ST_STU	ST_DEF	14	14	0,221	0,212	4.713.311	5.641.827	4.713.311
232	91511	VAR_A	ST_STU	ST_VWX	14	14	0,038	0,015	4.713.311	4.356.982	4.713.311
222	91511	VAR_A	ST_VWX	ST_ABC	43	31	0,019	0,012	5.595.674	5.780.019	5.873.983
224	91511	VAR_A	ST_VWX	ST_DEF	43	28	0,205	0,182	5.566.725	6.810.717	5.873.983
226	91511	VAR_A	ST_VWX	ST_STU	43	14	0,031	0,012	4.356.982	4.713.311	5.873.983

Sum of all scores within the aggregate *91511* for variable *VAR_A*, for the combination of the two indicated statistics.

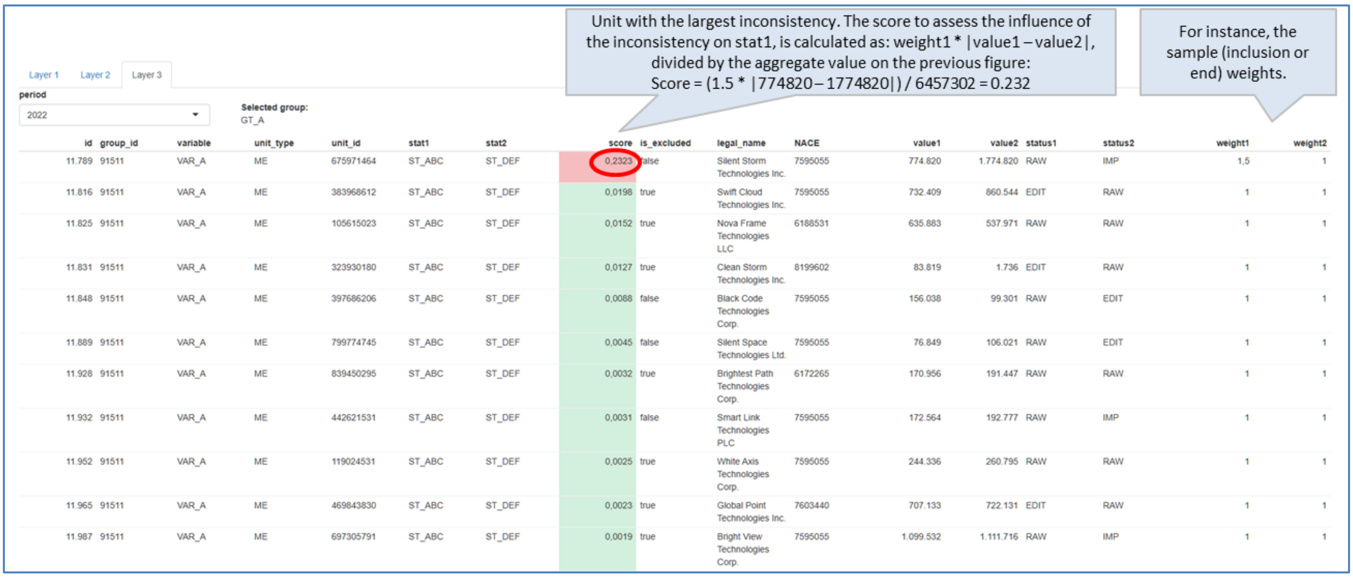
The largest inconsistency within the aggregate *91511* for variable *VAR_A*, occurs for the combination of the two statistics *ST_ABC* and *ST_DEF*.

Count and *overlap* indicate the number of units within the aggregate for stat1, and the number of overlapping units with stat2.

Aggregate values of *stat1* and *stat2* are computed based only on overlapping units. The weights from *stat1* are applied to assess the influence on that statistic.

Aggregate value that is used in the denominator of the score (**see next figure**). By default the aggregate value for stat1 (also non-overlapping units), but other stats can be used.

Figure 5. Dashboard: (output) aggregate - variable



Unit with the largest inconsistency. The score to assess the influence of the inconsistency on stat1, is calculated as: weight1 * |value1 – value2|, divided by the aggregate value on the previous figure:
Score = (1.5 * |774820 – 1774820|) / 6457302 = 0.232

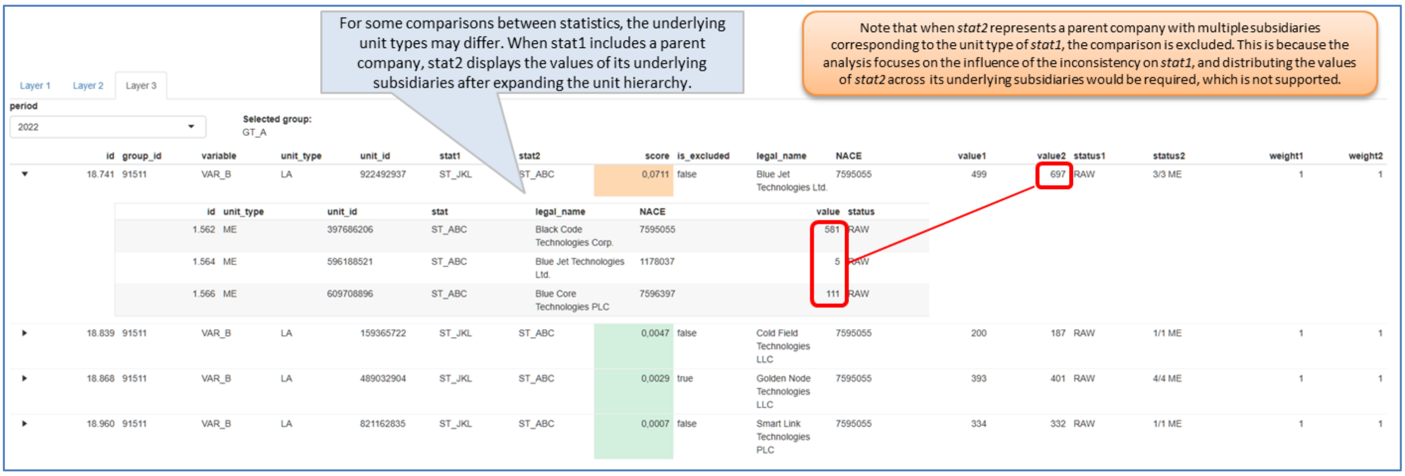
For instance, the sample (inclusion or end) weights.

period
2022

Selected group:
GT_A

id	group_id	variable	unit_type	unit_id	stat1	stat2	score	is_excluded	legal_name	NACE	value1	value2	status1	status2	weight1	weight2
11.789	91511	VAR_A	ME	675971464	ST_ABC	ST_DEF	0,2323	false	Silent Storm Technologies Inc.	7595055	774.820	1.774.820	RAW	IMP	1,5	1
11.816	91511	VAR_A	ME	383968612	ST_ABC	ST_DEF	0,0198	true	Swift Cloud Technologies Inc.	7595055	732.409	860.544	EDIT	RAW	1	1
11.825	91511	VAR_A	ME	105615023	ST_ABC	ST_DEF	0,0152	true	Nova Frame Technologies LLC	6188531	635.883	537.971	RAW	RAW	1	1
11.831	91511	VAR_A	ME	323930180	ST_ABC	ST_DEF	0,0127	true	Clean Storm Technologies Inc.	8199602	83.819	1.736	EDIT	RAW	1	1
11.848	91511	VAR_A	ME	397686206	ST_ABC	ST_DEF	0,0088	false	Black Code Technologies Corp.	7595055	156.038	99.301	RAW	EDIT	1	1
11.889	91511	VAR_A	ME	799774745	ST_ABC	ST_DEF	0,0045	false	Silent Space Technologies Ltd.	7595055	76.849	106.021	RAW	EDIT	1	1
11.928	91511	VAR_A	ME	839450295	ST_ABC	ST_DEF	0,0032	true	Brightest Path Technologies Corp.	6172265	170.956	191.447	RAW	RAW	1	1
11.932	91511	VAR_A	ME	442621531	ST_ABC	ST_DEF	0,0031	false	Smart Link Technologies PLC	7595055	172.564	192.777	RAW	IMP	1	1
11.952	91511	VAR_A	ME	119024531	ST_ABC	ST_DEF	0,0025	true	White Axis Technologies Corp.	7595055	244.336	260.795	RAW	RAW	1	1
11.965	91511	VAR_A	ME	469843830	ST_ABC	ST_DEF	0,0023	true	Global Point Technologies Inc.	7603440	707.133	722.131	EDIT	RAW	1	1
11.987	91511	VAR_A	ME	697305791	ST_ABC	ST_DEF	0,0019	true	Bright View Technologies Corp.	7595055	1.099.532	1.111.716	RAW	IMP	1	1

Figure 6. Dashboard: underlying units.



For some comparisons between statistics, the underlying unit types may differ. When stat1 includes a parent company, stat2 displays the values of its underlying subsidiaries after expanding the unit hierarchy.

Note that when *stat2* represents a parent company with multiple subsidiaries corresponding to the unit type of *stat1*, the comparison is excluded. This is because the analysis focuses on the influence of the inconsistency on *stat1*, and distributing the values of *stat2* across its underlying subsidiaries would be required, which is not supported.

period
2022

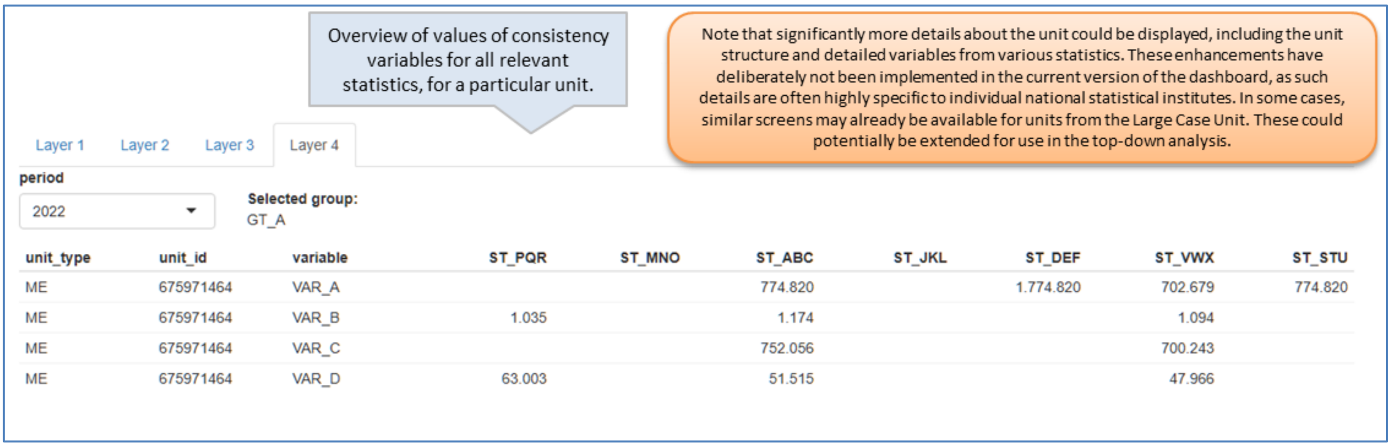
Selected group:
GT_A

	id	group_id	variable	unit_type	unit_id	stat1	stat2	score	is_excluded	legal_name	NACE	value1	value2	status1	status2	weight1	weight2
▼	18.741	91511	VAR_B	LA	922492937	ST_JKL	ST_ABC	0,0711	false	Blue Jet Technologies Ltd.	7595055	499	697	RAW	3/3 ME	1	1

id	unit_type	unit_id	stat	legal_name	NACE	value	status
1.562	ME	397686206	ST_ABC	Black Code Technologies Corp.	7595055	581	RAW
1.564	ME	596188521	ST_ABC	Blue Jet Technologies Ltd.	1178037	5	RAW
1.566	ME	609708896	ST_ABC	Blue Core Technologies PLC	7596397	111	RAW

	id	group_id	variable	unit_type	unit_id	stat1	stat2	score	is_excluded	legal_name	NACE	value1	value2	status1	status2	weight1	weight2
▶	18.839	91511	VAR_B	LA	159365722	ST_JKL	ST_ABC	0,0047	false	Cold Field Technologies LLC	7595055	200	187	RAW	1/1 ME	1	1
▶	18.868	91511	VAR_B	LA	489032904	ST_JKL	ST_ABC	0,0029	true	Golden Node Technologies LLC	7595055	393	401	RAW	4/4 ME	1	1
▶	18.960	91511	VAR_B	LA	821162835	ST_JKL	ST_ABC	0,0007	false	Smart Link Technologies PLC	7595055	334	332	RAW	1/1 ME	1	1

Figure 7. Dashboard: units including subsidiaries.



Overview of values of consistency variables for all relevant statistics, for a particular unit.

Note that significantly more details about the unit could be displayed, including the unit structure and detailed variables from various statistics. These enhancements have deliberately not been implemented in the current version of the dashboard, as such details are often highly specific to individual national statistical institutes. In some cases, similar screens may already be available for units from the Large Case Unit. These could potentially be extended for use in the top-down analysis.

period
2022

Selected group:
GT_A

unit_type	unit_id	variable	ST_PQR	ST_MNO	ST_ABC	ST_JKL	ST_DEF	ST_VWX	ST_STU
ME	675971464	VAR_A			774.820		1.774.820	702.679	774.820
ME	675971464	VAR_B	1.035		1.174			1.094	
ME	675971464	VAR_C			752.056			700.243	
ME	675971464	VAR_D	63.003		51.515			47.966	

Figure 8. Dashboard: overview per unit.

3.4 Discussion and conclusions

The goal of further enhancing the consistency dashboard – making it more flexible and widely applicable – has been achieved. A broad range of statistics and other data sources can now be compared within the dashboard, across different periods, variables, unit types and output aggregate types. This is made possible by simply including the relevant data and metadata in the standardized input and auxiliary tables.

A key lesson was the importance of separating data processing from presentation. This became clear after early challenges in comparing multiple data sources across different unit types and aggregation levels. These issues arose from using prototype code from a proof of concept (POC), which lacked the needed flexibility. The codebase was therefore fully refactored: data preparation and processing were rebuilt in Microsoft SQL Server to improve performance, scalability, and clarity, while the dashboard front end remains in R Shiny. A public version of the code developed under this grant is available at:
https://github.com/SNStatComp/ConsistencyDashboard.

Possible Improvements and Enhancements:

- Add absolute differences alongside relative scores for more intuitive interpretation;
- Allow users to set a focus on a single statistic, so that only combinations involving this focus statistic (as *stat1*) are included in the Layer 1 scores. The other layers remain unchanged. This focused view can be especially useful in the final stages before the publication of the selected statistic;
- Enable score denominators to be based on alternative variables, for example in cases where the main variable can take both positive and negative values (e.g., balances).

Some possible enhancements have deliberately not been implemented in the current version, as such details are often highly specific to individual national statistical institutes. However, for CBS, this functionality will be developed by integrating the dashboard into the new integrated uniform production system that is currently being built for all business statistics. For instance:

- The dashboard can be deployed on e.g. a Kubernetes cluster, allowing for scalable, containerized execution with integrated resource management and monitoring;
- Input table based on detailed variables, including a derivation scheme for consistency variables;
- Include a refresh button (or optionally trigger the refresh through an automated nightly job);
- In Layer 4, significantly more details about the unit can be displayed, including the unit structure and detailed variables from various statistics. This can be combined with an additional layer dedicated to Large Case Units (LCUs), which primarily focus on these detailed unit-level data. Integrating LCU-specific screens with the top-down analysis views used for non-LCU units enables the top-down workflow to directly benefit from the richer interface developed for LCUs, reducing duplication and enhancing consistency across user experience;
- Comparing raw and edited data within the dashboard. This is already possible using the current dashboard functionality, by treating raw and edited data as separate sources. If needed, a further distinction can be made between automatically and manually edited data. The dashboard can then be used to identify the largest adjustments, while Layer 2 also allows for the detection of more structural differences in cases where there are no major adjustments.

3.5 References

A. Vaasen-Otten, F. Aelen, S. Scholtus and W. de Jong (2022), Towards a New Integrated Uniform Production System for Business Statistics at Statistics Netherlands: Quality Indicators to Guide Top-down Analysis. UNECE Expert Meeting on Statistical Data Editing, 3-7 October 2022 (virtual).

Appendix A: Sample case of multisource editing

In this appendix, we provide a small fictional example to illustrate steps 3, 4 and 5 of the multisource editing procedure defined in Section 2.2.

Table 4. Example with two data sources and two common variables.

ID	common variables		data source 1					data source 2				
	z_1	z_2	$x_1^{(1)}$	$x_2^{(1)}$	$y_1^{(1)}$	$y_2^{(1)}$	$y_3^{(1)}$	$x_1^{(2)}$	$x_2^{(2)}$	$y_1^{(2)}$	$y_2^{(2)}$	$y_3^{(2)}$
1	.	.	100	0	100	0	0	800	160	800	160	960
2	.	.	600	800	600	300	500	690	800	690	800	1490

Table 4 contains fictional data on two units observed in two data sources. In this example, two common variables are available in both sources. The two data sources each contain three observed variables: $\mathbf{y}_i^{(1)} = \left(y_{i1}^{(1)}, y_{i2}^{(1)}, y_{i3}^{(1)}\right)'$ and $\mathbf{y}_i^{(2)} = \left(y_{i1}^{(2)}, y_{i2}^{(2)}, y_{i3}^{(2)}\right)'$. In the first source, the following internal edit rules (3) apply:

$$\begin{aligned} y_{i1}^{(1)} &\geq 0; \\ y_{i3}^{(1)} &\geq 0; \\ y_{i2}^{(1)} &\geq y_{i3}^{(1)}. \end{aligned} \tag{16}$$

Similarly, the following internal edit rules are relevant for the second data source:

$$\begin{aligned} y_{i1}^{(2)} &\geq 0; \\ y_{i2}^{(2)} &\geq 0; \\ y_{i3}^{(2)} &= y_{i1}^{(2)} + y_{i2}^{(2)}. \end{aligned} \tag{17}$$

The values of the common variables $\mathbf{x}_i^{(1)} = \left(x_{i1}^{(1)}, x_{i2}^{(1)}\right)$ and $\mathbf{x}_i^{(2)} = \left(x_{i1}^{(2)}, x_{i2}^{(2)}\right)$ in Table 4 were derived from the observed variables $\mathbf{y}_i^{(1)}$ and $\mathbf{y}_i^{(2)}$ by the following rules (4):

$$\begin{aligned} x_{i1}^{(1)} &= y_{i1}^{(1)}; \\ x_{i2}^{(1)} &= y_{i2}^{(1)} + y_{i3}^{(1)}; \end{aligned} \tag{18}$$

and

$$\begin{aligned} x_{i1}^{(2)} &= y_{i1}^{(2)}; \\ x_{i2}^{(2)} &= y_{i2}^{(2)}. \end{aligned} \tag{19}$$

The universal values of the common variables, $\mathbf{z}_i = (z_{i1}, z_{i2})'$ should satisfy the following edit rules (7):

$$\begin{gathered} z_{i1} \geq 0; \\ z_{i2} \geq 0; \\ \text{IF } (z_{i1} > 0) \text{ THEN } (z_{i2} > 0). \end{gathered} \tag{20}$$

Finally, we include edit rules of the form (11*) with $\varepsilon_1^* = \varepsilon_2^* = 0.05$ and $\Delta_1^* = \Delta_2^* = 50$ to relate the values of the common variables to their universal values. [Note that the universal values z_{i1} and z_{i2} are known to be non-negative by (20).] We obtain, for $p \in \{1,2\}$,

$$\begin{gathered} 0.95 z_{i1} - 50 \leq x_{i1}^{(p)} \leq 1.05 z_{i1} + 50; \\ 0.95 z_{i2} - 50 \leq x_{i2}^{(p)} \leq 1.05 z_{i2} + 50. \end{gathered} \tag{21}$$

To make the example more concrete, the following interpretation could be given to the variables:

- z_{i1}: purchasing value of goods;
- z_{i2}: total other costs;
- $y_{i1}^{(1)}$: purchasing value of goods according to source 1;
- $y_{i2}^{(1)}$: staff costs according to source 1;
- $y_{i3}^{(1)}$: miscellaneous costs according to source 1;
- $x_{i1}^{(1)}$: purchasing value of goods according to source 1 ($= y_{i1}^{(1)}$);
- $x_{i2}^{(1)}$: total other costs according to source 1 ($= y_{i2}^{(1)} + y_{i3}^{(1)}$);
- $y_{i1}^{(2)}$: purchasing value of goods according to source 2;
- $y_{i2}^{(2)}$: total other costs according to source 2;
- $y_{i3}^{(2)}$: total costs according to source 2.
- $x_{i1}^{(2)}$: purchasing value of goods according to source 2 ($= y_{i1}^{(2)}$);
- $x_{i2}^{(2)}$: total other costs according to source 2 ($= y_{i2}^{(2)}$).

In step 3 of the automatic editing procedure, we consider the values $\left(\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}, \mathbf{z}_i\right)$ and their edit rules (20) and (21). For the first record in Table 4, it is clear that the values $x_{i1}^{(1)} = 100$ and $x_{i1}^{(2)} = 800$ are too far apart given the restrictions in (21), and similarly $x_{i2}^{(1)} = 0$ and $x_{i2}^{(2)} = 160$ are too far apart as well. For both common variables, at least one of the observed values must be considered incorrect. For the second record, all values are close enough to be considered correct.

For step 3, a Fellegi-Holt-based error localization problem of the form (2) is set up for each record in Table 4. Suppose that the values of the common variables in the first source are considered a priori slightly more reliable than those in the second source. We reflect this by assigning a reliability weight of 2 to the values in $\mathbf{x}_i^{(1)}$ and a reliability weight of 1 to the values in $\mathbf{x}_i^{(2)}$. For the first record in Table 4, this yields the following MILP problem:

$$\begin{aligned} &\min\left(2\delta_{x1}^{(1)} + 2\delta_{x2}^{(1)} + \delta_{x1}^{(2)} + \delta_{x2}^{(2)}\right) \text{ under the following restrictions:} \\ &\quad \left(\tilde{x}_{i1}^{(1)}, \tilde{x}_{i2}^{(1)}, \tilde{x}_{i1}^{(2)}, \tilde{x}_{i2}^{(2)}, \tilde{z}_{i1}, \tilde{z}_{i2}\right)' \text{ satisfies all edit rules (20) and (21);} \\ &\quad 100 - M\delta_{x1}^{(1)} \le \tilde{x}_{i1}^{(1)} \le 100 + M\delta_{x1}^{(1)}; \\ &\quad 0 - M\delta_{x2}^{(1)} \le \tilde{x}_{i2}^{(1)} \le 0 + M\delta_{x2}^{(1)}; \\ &\quad 800 - M\delta_{x1}^{(2)} \le \tilde{x}_{i1}^{(2)} \le 800 + M\delta_{x1}^{(2)}; \\ &\quad 160 - M\delta_{x2}^{(2)} \le \tilde{x}_{i2}^{(2)} \le 160 + M\delta_{x2}^{(2)}; \\ &\quad \left(\delta_{x1}^{(1)}, \delta_{x2}^{(1)}, \delta_{x1}^{(2)}, \delta_{x2}^{(2)}\right)' \in \{0,1\}^4. \end{aligned}$$

The optimal solution to this problem is $\left(\delta_{x1}^{(1)}, \delta_{x2}^{(1)}, \delta_{x1}^{(2)}, \delta_{x2}^{(2)}\right) = (0,1,1,0)$, with a total weight of 3. Under this solution, it is decided to change $x_{i1}^{(2)}$ for the first common variable and $x_{i2}^{(1)}$ for the second common variable. For the second record in Table 4, it is found that the original values are already consistent with all edit rules in (20) and (21) – i.e., it is possible to find values for $\mathbf{z}_i$ that satisfy these edit rules together with $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_i^{(2)}$ – so here no values are considered erroneous. Table 5 shows the edited data after step 3.

Table 5. Edited data after step 3.

ID	**common variables**		**data source 1**					**data source 2**				
	z_1	z_2	$x_1^{(1)}$	$x_2^{(1)}$	$y_1^{(1)}$	$y_2^{(1)}$	$y_3^{(1)}$	$x_1^{(2)}$	$x_2^{(2)}$	$y_1^{(2)}$	$y_2^{(2)}$	$y_3^{(2)}$
1	.	.	100	.	100	0	0	.	160	800	160	960
2	.	.	600	800	600	300	500	690	800	690	800	1490

In step 4, we begin by imputing the universal values of z_{i1} and z_{i2}. For this small example, it suffices to use a simple ad hoc procedure which uses only the first two options from the general imputation strategy proposed in Section 2.5.4:

- If any of the observed values $x_{il}^{(p)}$ are not missing after step 3 and do not cause violations of edit rules (20) and (21), then impute such a value. If multiple values are available, then choose the one with the largest reliability weight from step 3.
- Otherwise, impute the midpoint of the feasible interval for z_{il}, given edit rules (20) and (21).

The resulting data for the example are shown in Table 6. Note that in the second record, $x_{i1}^{(1)} = 600$ and $x_{i1}^{(2)} = 690$ are themselves not feasible values for z_{i1}. The actual feasible interval for z_{i1} for this record is:

$$\left[\frac{690-50}{1.05}, \frac{600+50}{0.95}\right] \approx [609.5, 684.2].$$

The midpoint of this interval is used to impute z_{i1} in the second record.

Table 6. Edited data after step 4.

ID	common variables		data source 1					data source 2				
	z_1	z_2	$x_1^{(1)}$	$x_2^{(1)}$	$y_1^{(1)}$	$y_2^{(1)}$	$y_3^{(1)}$	$x_1^{(2)}$	$x_2^{(2)}$	$y_1^{(2)}$	$y_2^{(2)}$	$y_3^{(2)}$
1	**100**	**160**	100	.	100	0	0	.	160	800	160	960
2	**646.9**	**800**	600	800	600	300	500	690	800	690	800	1490

In the second part of step 4, we derive new edit rules for $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_i^{(2)}$ by substituting the imputed values of z_{i1} and z_{i2} in (21). For the first record, this yields

$$\begin{gathered} 45 \leq x_{i1}^{(p)} \leq 155; \\ 102 \leq x_{i2}^{(p)} \leq 218; \end{gathered} \tag{22}$$

and for the second record

$$\begin{gathered} 564.5 \leq x_{i1}^{(p)} \leq 729.2; \\ 710 \leq x_{i2}^{(p)} \leq 890. \end{gathered} \tag{23}$$

In step 5, the data in each source are edited separately. Again, we suppose no deductive correction rules have been specified. For each record in each data source, an error localization problem of the form (2) is set up. For simplicity, suppose all reliability weights are chosen equal to 1. As an example, for the first record in data source 1 in Table 6, we obtain the following MILP problem:

$$\begin{aligned} &\min\left(\delta_{x1}^{(1)} + \delta_{y1}^{(1)} + \delta_{y2}^{(1)} + \delta_{y3}^{(1)}\right) \text{ under the following restrictions:} \\ &\quad \left(\tilde{x}_{i1}^{(1)}, \tilde{x}_{i2}^{(1)}, \tilde{y}_{i1}^{(1)}, \tilde{y}_{i2}^{(1)}, \tilde{y}_{i3}^{(1)}\right)' \text{ satisfies all edit rules (16), (18) and (22);} \\ &\quad 100 - M\delta_{x1}^{(1)} \leq \tilde{x}_{i1}^{(1)} \leq 100 + M\delta_{x1}^{(1)}; \\ &\quad 100 - M\delta_{y1}^{(1)} \leq \tilde{y}_{i1}^{(1)} \leq 100 + M\delta_{y1}^{(1)}; \\ &\quad 0 - M\delta_{y2}^{(1)} \leq \tilde{y}_{i2}^{(1)} \leq 0 + M\delta_{y2}^{(1)}; \\ &\quad 0 - M\delta_{y3}^{(1)} \leq \tilde{y}_{i3}^{(1)} \leq 0 + M\delta_{y3}^{(1)}; \\ &\quad \left(\delta_{x1}^{(1)}, \delta_{y1}^{(1)}, \delta_{y2}^{(1)}, \delta_{y3}^{(1)}\right)' \in \{0,1\}^4. \end{aligned}$$

The optimal solution to this problem is to change only the value of $y_{i2}^{(1)}$, with a total weight of 1.

Table 7 shows the edited data after error localization for both records in both data sources. Additional values in $\mathbf{y}_i^{(1)}$ and $\mathbf{y}_i^{(2)}$ were identified as erroneous. In record 1, this was done to accommodate errors in the common variables $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_i^{(2)}$ that were found in step 3, given the relations in (18) and (19). In record 2, this was done to resolve an inconsistency with respect to the internal edit rules (16) in data source 1. Finally, Table 8 shows a possible way to impute the missing values in Table 7 that is consistent with all restrictions. Note that because of edit rules (22) and (23), no new inconsistencies between data sources were introduced during step 5, even though each data source was edited independently.

Table 7. Edited data after error localization in step 5.

ID	**common variables**		**data source 1**					**data source 2**				
	z_1	z_2	$x_1^{(1)}$	$x_2^{(1)}$	$y_1^{(1)}$	$y_2^{(1)}$	$y_3^{(1)}$	$x_1^{(2)}$	$x_2^{(2)}$	$y_1^{(2)}$	$y_2^{(2)}$	$y_3^{(2)}$
1	**100**	**160**	100	**.**	100	**.**	0	**.**	160	**.**	160	**.**
2	**646.9**	**800**	600	800	600	**.**	**.**	690	800	690	800	1490

Table 8. Final edited data after step 5.

ID	**common variables**		**data source 1**					**data source 2**				
	z_1	z_2	$x_1^{(1)}$	$x_2^{(1)}$	$y_1^{(1)}$	$y_2^{(1)}$	$y_3^{(1)}$	$x_1^{(2)}$	$x_2^{(2)}$	$y_1^{(2)}$	$y_2^{(2)}$	$y_3^{(2)}$
1	**100**	**160**	100	**160**	100	**160**	0	**100**	160	**100**	160	**260**
2	**646.9**	**800**	600	800	600	**500**	**300**	690	800	690	800	1490

Appendix B: Overview of common variables per data source

	Source								
Common variable	SBS	Prod Com	SFLE	STSA	STSS	SES	SIGS	PD	IS
Net turnover minus excise duties	X			X	X			X	
Industrial turnover minus excise duties	X	X			X				
Trade and other turnover	X	X			X				
Net turnover minus excise duties minus allocated freight costs	X	X							
Self-manufactured turnover in the Netherlands minus excise duties and allocated freight costs	X	X							
Turnover of Industrial Services	X	X							
Production	X		X						
Subsidies	X		X						
Average number of employees in fte	X		X			X			
Labor costs	X		X					X	
Wages	X					X		X	
Social costs	X							X	
Pension costs	X							X	
Other personnel costs	X							X	
Outsourced work	X							X	
Purchasing value	X							X	
Purchasing value except for outsourced work	X							X	
Sales margin	X							X	
Balance of book profits and losses on sales	X		X						
Net extraordinary gains and losses	X		X						
Operating profit	X		X						
Depreciation excluding that on operational leases	X		X						
Depreciation on operational leases	X		X						
Reversals of Impairments	X		X						
Earnings before interest, taxes, depreciation, and amortization (EBITDA)	X		X						
Financial result on lease	X		X						
Import			X				X		
Export			X				X		
Investments in physical fixed assets (PD definition)								X	X
Investments in physical fixed assets (SFLE definition)			X						X
Investments in intangible fixed assets (PD definition)								X	X
Investments in intangible fixed assets (SFLE definition)			X						X
Investments in leased assets			X						X