UNITED NATIONS ECONOMIC COMMISSION FOR EUROPE
CONFERENCE OF EUROPEAN STATISTICIANS
Expert Meeting on Statistical Data Editing
7-9 October 2024, Vienna

Current work on automatic multisource editing at Statistics Netherlands

Sander Scholtus, Arnout van Delden, Rob Willems, Frank Aelen (Statistics Netherlands, the Netherlands)
s.scholtus@cbs.nl

This work is supported by Eurostat grant SMP-ESS-2023-EBS-IBA. The views expressed in this paper are those of the authors and do not necessarily reflect the policies of Statistics Netherlands.

I. Introduction

1. As part of a new integrated uniform production system for business statistics, Statistics Netherlands aims to develop a system for simultaneous editing of variables that are observed across different data sources. In this way inconsistencies between statistics could be identified and resolved as early as possible, which should increase overall data quality and is considered more efficient and effective than the current approach, where data for different statistics are edited mostly in isolation. (An exception is the so-called Large Cases Unit which already performs manual integrated editing for the largest and most complicated enterprise groups.) As a result, sometimes large inconsistencies between statistics are found at a late stage, e.g., during the production of National Accounts. The new approach will involve top-down interactive editing across statistics, using score functions to identify the most influential inconsistencies. We refer to Vaasen-Otten et al. (2022) for a discussion of top-down multisource editing and of the wider context of the new production system.

2. In addition to top-down interactive editing, we aim to introduce automatic editing of other inconsistencies across data sources where possible. An initial approach and results for automatic multisource editing from a pilot study were previously presented in Scholtus et al. (2022). In the present paper, we will give an update of the work that is being done in this area. Currently, there is a project at Statistics Netherlands to develop automatic multisource editing further for eventual use in regular statistical production. This project is supported as part of a grant from Eurostat for timelier, more relevant and more integrated European business statistics. The project will run from January 2024 until June 2025.

3. Regarding automatic editing across statistics, the main aims of the project are, first, to improve the quality of automatically edited data by introducing better edit rules and incorporating more unit-specific information, and second, to evaluate the effects of automatic editing on the quality of statistical output. While the project is still ongoing, we will present some ideas and initial developments. The paper is organized as follows. Section II provides a summary of the automatic editing methodology used in this project, including a toy example to illustrate the various process steps. Sections III and IV discuss ideas for improving and evaluating the quality of automatic editing, respectively.

II. Methodology

A. Automatic editing methods

4. 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).

5. 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 $\mathbf{x} = (x_1, \dots, x_J)'$ 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:

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

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 $\leq$ may also be replaced by $\geq$, $<$, $>$ or $=$. A special case of an edit rule of the form (1) is a simple linear inequality $\mathbf{a}'\mathbf{x} \leq b$ 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).]

6. 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):

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

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 $-M \leq \tilde{x}_j - x_j \leq M$ when $\delta_j = 1$ (i.e., in practical terms any adjustment can be made to the original value). In practice so far – and in the toy example that will be discussed below – 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).

7. 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.

8. 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. 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).

B. Multisource editing: notation and setup

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

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

10. 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 $y_{il}^{(p)}$ 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 $\mathbf{y}_i^{(p)}$ denote a vector containing all $y_{il}^{(p)}$ that occur in data source p. In general, a common variable may not be observed directly in a source but has to be derived from the observed variables $\mathbf{x}_i^{(p)}$. We assume here that all derivations are affine transformations, so it holds that

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

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

11. 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 values of common variables in different data sources, we define further restrictions of the following form:

$$\left|y_{il}^{(p)} - y_{il}^{(q)}\right| \leq \varepsilon_l \left|y_{il}^{(q)}\right| \tag{5}$$

for all pairs (p, q) with $p \in B_{il}$, $q \in B_{il}$. Here, the parameter $0 \leq \varepsilon_l < 1$ defines the maximal allowed relative deviation between observed values of common variable l. So far, we have used $\varepsilon_l = 0.1$ (i.e., relative deviations of up to 10%). Choosing $\varepsilon_l = 0$ 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.

12. To formulate the automatic multisource editing problem, it is useful to introduce a vector of 'true' values of the common variables for unit i, $\mathbf{z}_i = (z_{i1}, \dots, z_{iL})'$. Instead of (5), we may then define the restrictions

$$\left|y_{il}^{(p)} - z_{il}\right| \leq \varepsilon_l^* |z_{il}| \tag{6}$$

for all $p \in B_{il}$. With the choice $\varepsilon_l^* = \varepsilon_l/(2+\varepsilon_l)$, it can be shown using the triangle inequality that any set of values that satisfies all restrictions (6) also satisfies all restrictions (5) (Scholtus et al., 2022). It should be noted that each restriction (6) can be written as a set of edit rules of the form (1):

$$\begin{aligned} &\text{IF } (z_{il} \geq 0) \text{ THEN } \left(y_{il}^{(p)} \geq (1-\varepsilon_l^*) z_{il}\right); \\ &\text{IF } (z_{il} \geq 0) \text{ THEN } \left(y_{il}^{(p)} \leq (1+\varepsilon_l^*) z_{il}\right); \\ &\text{IF } (z_{il} < 0) \text{ THEN } \left(y_{il}^{(p)} \geq (1+\varepsilon_l^*) z_{il}\right); \\ &\text{IF } (z_{il} < 0) \text{ THEN } \left(y_{il}^{(p)} \leq (1-\varepsilon_l^*) z_{il}\right). \end{aligned} \tag{6*}$$

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

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

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

13. In brief, the purpose of automatic multisource editing is to obtain data for unit i, consisting of $\left(\mathbf{x}_i^{(1)}, \ldots, \mathbf{x}_i^{(P)}, \mathbf{y}_i^{(1)}, \ldots, \mathbf{y}_i^{(P)}, \mathbf{z}_i\right)$, 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, they proposed a three-step procedure:

1) Automatic editing of common variables across data sources
   In step 1, errors are identified in the common variables $\left(\mathbf{y}_i^{(1)}, \ldots, \mathbf{y}_i^{(P)}, \mathbf{z}_i\right)$, using edit rules (6*) and (7).
2) Imputing 'true' values and deriving additional edit rules for the common variables
   Using the edited data from step 1, the 'true' 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 $\left(\mathbf{y}_i^{(1)}, \ldots, \mathbf{y}_i^{(P)}\right)$.
3) Automatic editing within each individual data source
   Step 3 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 values of $\mathbf{z}_i$ from step 2 may not be edited during this step.

14. In step 1 and step 3, both deductive correction and error localization [or indeed any other automatic editing method that can account for edit rules of the form (3), (4), (6*) and (7)] could be applied. Here, we will illustrate the procedure using a toy example. A detailed description of the three steps, as well as a larger, more realistic example, can be found in Scholtus et al. (2022).

C. Example

Table 1. Example with two data sources and two common variables.

ID	common variables		data source 1					data source 2				
	z_1	z_2	$y_1^{(1)}$	$y_2^{(1)}$	$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$	$y_1^{(2)}$	$y_2^{(2)}$	$x_1^{(2)}$	$x_2^{(2)}$	$x_3^{(2)}$
1	.	.	100	0	100	0	0	800	160	800	160	960
2	.	.	60	80	60	30	50	65	80	65	80	145

15. Table 1 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: $\mathbf{x}_i^{(1)} = \left(x_{i1}^{(1)}, x_{i2}^{(1)}, x_{i3}^{(1)}\right)'$ and $\mathbf{x}_i^{(2)} = \left(x_{i1}^{(2)}, x_{i2}^{(2)}, x_{i3}^{(2)}\right)'$. In the first source, the following internal edit rules (3) apply:

$$\begin{aligned} &x_{i1}^{(1)} \geq 0; \\ &x_{i3}^{(1)} \geq 0; \\ &x_{i2}^{(1)} \geq x_{i3}^{(1)}. \end{aligned} \tag{8}$$

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

$$\begin{aligned} x_{i1}^{(2)} &\geq 0; \\ x_{i2}^{(2)} &\geq 0; \\ x_{i3}^{(2)} &= x_{i1}^{(2)} + x_{i2}^{(2)}. \end{aligned} \tag{9}$$

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

$$\begin{aligned} y_{i1}^{(1)} &= x_{i1}^{(1)}; \\ y_{i2}^{(1)} &= x_{i2}^{(1)} + x_{i3}^{(1)}; \end{aligned} \tag{10}$$

and

$$\begin{aligned} y_{i1}^{(2)} &= x_{i1}^{(2)}; \\ y_{i2}^{(2)} &= x_{i2}^{(2)}. \end{aligned} \tag{11}$$

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

$$\begin{aligned} z_{i1} &\geq 0; \\ z_{i2} &\geq 0; \\ \text{IF } (z_{i1} > 0) \text{ THEN } &(z_{i2} > 0). \end{aligned} \tag{12}$$

Finally, we include edit rules of the form (6) or (6*) with $\varepsilon_1^* = \varepsilon_2^* = 0.05$ to relate the values of the common variables to their 'true' values. Since the 'true' values z_{i1} and z_{i2} are known to be non-negative by (12), in this case these edit rules can be reduced to a simpler form: for $p \in \{1,2\}$,

$$\begin{aligned} 0.95 z_{i1} \leq y_{i1}^{(p)} \leq 1.05 z_{i1}; \\ 0.95 z_{i2} \leq y_{i2}^{(p)} \leq 1.05 z_{i2}. \end{aligned} \tag{13}$$

17. In step 1 of the automatic editing procedure, we consider the values $\left(\mathbf{y}_i^{(1)}, \mathbf{y}_i^{(2)}, \mathbf{z}_i\right)$ and their edit rules (12) and (13). For the first record in Table 1, it is clear that the values $y_{i1}^{(1)} = 100$ and $y_{i1}^{(2)} = 800$ are too far apart given the restrictions in (13), and similarly $y_{i2}^{(1)} = 0$ and $y_{i2}^{(2)} = 160$ 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.

18. We assume here that no deductive correction rules are specified for step 1. A Fellegi-Holt-based error localization problem of the form (2) is set up for each record in Table 1. 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 $\mathbf{y}_i^{(1)}$ and a reliability weight of 1 to the values in $\mathbf{y}_i^{(2)}$. For the first record in Table 1, this yields the following MILP problem:

$$\min\left(2\delta_{y1}^{(1)} + 2\delta_{y2}^{(1)} + \delta_{y1}^{(2)} + \delta_{y2}^{(2)}\right) \text{ under the following restrictions:}$$

$$\left(\tilde{y}_{i1}^{(1)}, \tilde{y}_{i2}^{(1)}, \tilde{y}_{i1}^{(2)}, \tilde{y}_{i2}^{(2)}, \tilde{z}_{i1}, \tilde{z}_{i2}\right)' \text{ satisfies all edit rules (12) and (13);}$$

$$100 - M\delta_{y1}^{(1)} \leq \tilde{y}_{i1}^{(1)} \leq 100 + M\delta_{y1}^{(1)};$$

$$0 - M\delta_{y2}^{(1)} \leq \tilde{y}_{i2}^{(1)} \leq 0 + M\delta_{y2}^{(1)};$$

$$800 - M\delta_{y1}^{(2)} \leq \tilde{y}_{i1}^{(2)} \leq 800 + M\delta_{y1}^{(2)};$$

$$160 - M\delta_{y2}^{(2)} \leq \tilde{y}_{i2}^{(2)} \leq 160 + M\delta_{y2}^{(2)};$$

$$\left(\delta_{y1}^{(1)}, \delta_{y2}^{(1)}, \delta_{y1}^{(2)}, \delta_{y2}^{(2)}\right)' \in \{0,1\}^4.$$

The optimal solution to this problem is $\left(\delta_{y1}^{(1)}, \delta_{y2}^{(1)}, \delta_{y1}^{(2)}, \delta_{y2}^{(2)}\right) = (0,1,1,0)$, with a total weight of 3. Under this solution, it is decided to change $y_{i1}^{(2)}$ for the first common variable and $y_{i2}^{(1)}$ for the second common variable. For the second record in Table 1, it is found that the original values are already consistent with all edit rules in (12) and (13) – i.e., it is possible to find values for $\mathbf{z}_i$ that satisfy these edit rules together with $\mathbf{y}_i^{(1)}$ and $\mathbf{y}_i^{(2)}$ – so here no values are considered erroneous. Table 2 shows the edited data after step 1.

Table 2. Edited data after step 1.

ID	common variables		data source 1					data source 2				
	z_1	z_2	$y_1^{(1)}$	$y_2^{(1)}$	$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$	$y_1^{(2)}$	$y_2^{(2)}$	$x_1^{(2)}$	$x_2^{(2)}$	$x_3^{(2)}$
1	.	.	100	.	100	0	0	.	160	800	160	960
2	.	.	60	80	60	30	50	65	80	65	80	145

19. In step 2, we begin by imputing the 'true' values of z_{i1} and z_{i2}. For this small example, it suffices to use a simple ad hoc procedure which fills in each z_{il} sequentially by the following rules:

- If any of the observed values $y_{il}^{(p)}$ are not missing after step 1 and do not cause violations of edit rules (12) and (13), then impute such a value. If multiple values are available, then choose the one with the largest reliability weight from step 1.
- Otherwise, impute the midpoint of the feasible interval for z_{il}, given edit rules (12) and (13).

For larger applications, such a sequential procedure does not always work. In general, we propose to use a sequential procedure to obtain initial imputations for z_{il} and then minimally adjust these imputations if necessary to satisfy all edit rules (12) and (13). (Recall that Fellegi-Holt-based error localization guarantees that a set of values always exists for $\mathbf{z}_i$ that satisfies all edit rules.) The resulting data for the example are shown in Table 3. Note that in the second record, $y_{i1}^{(1)} = 60$ and $y_{i1}^{(2)} = 65$ are themselves not feasible values for z_{i1}. The actual feasible interval for z_{i1} for this record is $[60/0.95,\ 65/1.05] \approx [61.90, 63.16]$.

Table 3. Edited data after step 2.

ID	common variables		data source 1					data source 2				
	z_1	z_2	$y_1^{(1)}$	$y_2^{(1)}$	$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$	$y_1^{(2)}$	$y_2^{(2)}$	$x_1^{(2)}$	$x_2^{(2)}$	$x_3^{(2)}$
1	**100**	**160**	100	.	100	0	0	.	160	800	160	960
2	**62.53**	**80**	60	80	60	30	50	65	80	65	80	145

20. In the second part of step 2, we derive new edit rules for $\mathbf{y}_i^{(1)}$ and $\mathbf{y}_i^{(2)}$ by substituting the imputed values of z_{i1} and z_{i2} in (13). For the first record, this yields

$$\begin{aligned} 95 \leq y_{i1}^{(p)} \leq 105; \\ 152 \leq y_{i2}^{(p)} \leq 168; \end{aligned} \tag{14}$$

and for the second record

$$\begin{aligned} 59.40 \leq y_{i1}^{(p)} \leq 65.66; \\ 76 \leq y_{i2}^{(p)} \leq 84. \end{aligned} \tag{15}$$

21. In step 3, 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 3, we obtain the following MILP problem:

$\min\left(\delta_{y1}^{(1)} + \delta_{x1}^{(1)} + \delta_{x2}^{(1)} + \delta_{x3}^{(1)}\right)$ under the following restrictions:

$\left(\tilde{y}_{i1}^{(1)}, \tilde{y}_{i2}^{(1)}, \tilde{x}_{i1}^{(1)}, \tilde{x}_{i2}^{(1)}, \tilde{x}_{i3}^{(1)}\right)'$ satisfies all edit rules (8), (10) and (14);

$100 - M\delta_{y1}^{(1)} \leq \tilde{y}_{i1}^{(1)} \leq 100 + M\delta_{y1}^{(1)}$;

$100 - M\delta_{x1}^{(1)} \leq \tilde{x}_{i1}^{(1)} \leq 100 + M\delta_{x1}^{(1)}$;

$$0 - M\delta_{x2}^{(1)} \leq \tilde{x}_{i2}^{(1)} \leq 0 + M\delta_{x2}^{(1)};$$
$$0 - M\delta_{x3}^{(1)} \leq \tilde{x}_{i3}^{(1)} \leq 0 + M\delta_{x3}^{(1)};$$
$$\left(\delta_{y1}^{(1)}, \delta_{x1}^{(1)}, \delta_{x2}^{(1)}, \delta_{x3}^{(1)}\right)' \in \{0,1\}^4.$$

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

22. Table 4 shows the edited data after error localization for both records in both data sources. Additional values in $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_i^{(2)}$ were identified as erroneous. In record 1, this was done to accommodate errors in the common variables $\mathbf{y}_i^{(1)}$ and $\mathbf{y}_i^{(2)}$ that were found in step 1, given the relations in (10) and (11). In record 2, this was done to resolve an inconsistency with respect to the internal edit rules (8) in data source 1. Finally, Table 5 shows a possible way to impute the missing values in Table 4 that is consistent with all restrictions. Note that because of edit rules (14) and (15), no new inconsistencies between data sources were introduced during step 3, even though each data source was edited independently.

Table 4. Edited data after error localization in step 3.

ID	common variables		data source 1					data source 2				
	z_1	z_2	$y_1^{(1)}$	$y_2^{(1)}$	$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$	$y_1^{(2)}$	$y_2^{(2)}$	$x_1^{(2)}$	$x_2^{(2)}$	$x_3^{(2)}$
1	100	160	100	.	100	.	0	.	160	.	160	.
2	62.53	80	60	80	60	.	.	65	80	65	80	145

Table 5. Final edited data after step 3.

ID	common variables		data source 1					data source 2				
	z_1	z_2	$y_1^{(1)}$	$y_2^{(1)}$	$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$	$y_1^{(2)}$	$y_2^{(2)}$	$x_1^{(2)}$	$x_2^{(2)}$	$x_3^{(2)}$
1	100	160	100	**160**	100	**160**	0	**100**	160	**100**	160	**260**
2	62.53	80	60	80	60	**50**	**30**	65	80	65	80	145

D. Implementation and pilot studies

23. A prototype implementation of the three-step procedure for automatic multisource editing has been 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 (Van der Loo and De Jonge, 2018 and 2021). An 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 three-step approach is computationally feasible but that the quality of edited data is 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.

24. The current project includes a new, larger pilot study with $P = 9$ data sources:

- Structural Business Statistics (survey)
- ProdCom (survey)
- Statistics on Finances of Large Enterprise groups (survey)
- Short-Term Statistics (admin data)
- Short-Term Statistics (survey)
- Statistics on Employees and Salaries (admin data)
- Statistics on International Trade of Goods and Services (combination of survey and admin data)
- Profit Declaration Tax Data (admin data)
- Investment Statistics (survey)

In total, $L = 33$ common variables have been identified. Of these, 27 variables occur in exactly two sources, five variables occur in exactly three sources, and one variable ('net turnover excluding excises') occurs in four sources. All data refer to the year 2022. In addition, production-edited data for the year 2021 are available as reference data. For step 1 and 2 of the editing procedure, all data sources will be considered. For step 3, we will initially focus on editing the Structural Business Statistics, where the current production process already includes extensive automatic editing.

III. Incorporating subject-matter knowledge into automatic editing

25. An important aim of the current project is to develop ways to take more subject-matter knowledge into account during automatic editing of the pilot study data. Three types of input that affect the outcome of automatic editing are: (*i*) deductive correction rules; (*ii*) edit rules for error localization; (*iii*) reliability weights for error localization. In this paper, we will focus on the latter two points.

A. Finding relevant edit rules

26. As explained in Section II, the multisource error localization problem in its current form involves edit rules of the forms (3), (4), (6*) and (7). Internal edit rules (3) for most data sources are already well-developed as part of regular statistical production. Exceptions may occur, e.g., for administrative data that are not yet used directly to create statistical output; in the pilot study this is true for Profit Declaration Tax Data. Edit rules (4) relating $\mathbf{y}_i^{(p)}$ to $\mathbf{x}_i^{(p)}$ are given by definition and edit rules (6*) follow immediately from the choice of ε_l^*. By contrast, edit rules (7) for the 'true' values of the common variables are still mostly lacking. Thus, finding edit rules for these variables seems to be a good opportunity for improvement.

27. 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 experience of the Large Cases Unit is 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. For now, we will use a more data-driven approach to find relations between common variables that can be turned into edit rules. The findings of these data analyses will also be discussed with subject-matter experts which, hopefully, can lead to even more edit rules being discovered.

28. As a starting point, we can take historical (internally) edited values $y_{il}^{(p)}$ in one particular data source p as a proxy for the underlying 'true' values z_{il}, and use these data to study patterns among a subset of the common variables. One, relatively simple, approach to find edit rules is to fit a linear regression model to each pair of variables $\left(y_{il_1}^{(p)}, y_{il_2}^{(p)}\right)$, with $y_{il_1}^{(p)}$ acting as independent variable and $y_{il_2}^{(p)}$ as dependent variable. For economic data, a regression model with heteroscedastic disturbances (variance proportional to the independent variable) often fits better than a model with homoscedastic disturbances. Next, we restrict attention to those pairs of variables where the explained variance of the linear model is large (R^2 greater than some threshold) and subject-matter experts consider the relation to be relevant. For each of these combinations of variables, the fitted regression model is used to obtain (e.g.) 95% prediction intervals for $y_{il_2}^{(p)}$ given $y_{il_1}^{(p)}$. The upper and lower bounds of these prediction intervals vary as a non-linear function of $y_{il_1}^{(p)}$ 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 $y_{il_2}^{(p)}$ given $y_{il_1}^{(p)}$, leading to a linear edit rule for z_{il_1} and z_{il_2} of the form:

$$\hat{\alpha}^{(\text{lower})} + \hat{\beta}^{(\text{lower})} z_{il_1} \leq z_{il_2} \leq \hat{\alpha}^{(\text{upper})} + \hat{\beta}^{(\text{upper})} z_{il_1}. \tag{16}$$

29. Figure 1 illustrates this approach. The solid blue line indicates the original fitted linear regression line. 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 coloured 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.

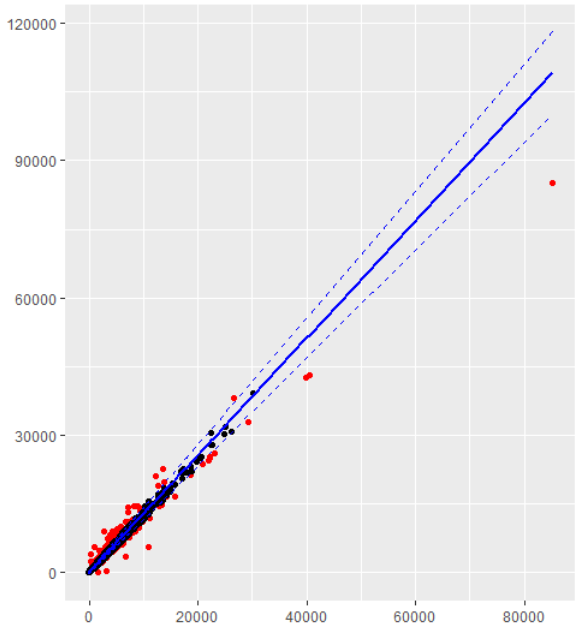

Figure 1. An illustration of the prediction-interval approach.

30. It should be noted that (16) 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 may be useful to force any adjustments made during automatic editing to satisfy this restriction, to avoid creating implausible combinations of values. To this end, the following variations on (16) could be used instead, for each data source p that includes common variable l_1 and/or l_2:

$$\begin{aligned}
&\text{IF } \left(y_{il_1}^{(p)} > a_{il_1}^{(p)}\right) \text{ THEN } \left(\hat{\alpha}^{(\text{lower})} + \hat{\beta}^{(\text{lower})} z_{il_1} \leq z_{il_2} \leq \hat{\alpha}^{(\text{upper})} + \hat{\beta}^{(\text{upper})} z_{il_1}\right);\\
&\text{IF } \left(y_{il_1}^{(p)} < a_{il_1}^{(p)}\right) \text{ THEN } \left(\hat{\alpha}^{(\text{lower})} + \hat{\beta}^{(\text{lower})} z_{il_1} \leq z_{il_2} \leq \hat{\alpha}^{(\text{upper})} + \hat{\beta}^{(\text{upper})} z_{il_1}\right);\\
&\text{IF } \left(y_{il_2}^{(p)} > a_{il_2}^{(p)}\right) \text{ THEN } \left(\hat{\alpha}^{(\text{lower})} + \hat{\beta}^{(\text{lower})} z_{il_1} \leq z_{il_2} \leq \hat{\alpha}^{(\text{upper})} + \hat{\beta}^{(\text{upper})} z_{il_1}\right);\\
&\text{IF } \left(y_{il_2}^{(p)} < a_{il_2}^{(p)}\right) \text{ THEN } \left(\hat{\alpha}^{(\text{lower})} + \hat{\beta}^{(\text{lower})} z_{il_1} \leq z_{il_2} \leq \hat{\alpha}^{(\text{upper})} + \hat{\beta}^{(\text{upper})} z_{il_1}\right).
\end{aligned} \tag{17}$$

Here, $a_{il_1}^{(p)}$ and $a_{il_2}^{(p)}$ denote the observed values of $y_{il_1}^{(p)}$ and $y_{il_2}^{(p)}$ in the original data, so initially it holds that $y_{il_1}^{(p)} = a_{il_1}^{(p)}$ and $y_{il_2}^{(p)} = a_{il_2}^{(p)}$. The restrictions in (17) imply that if any changes are made to $y_{il_1}^{(p)}$ or $y_{il_2}^{(p)}$ during automatic editing, then the values of z_{il_1} and z_{il_2} after editing have to conform to the bounds from the prediction-interval approach. If no changes are made to any $y_{il_1}^{(p)}$ or $y_{il_2}^{(p)}$, then these bounds do not apply.

31. It is necessary to check whether the assumption that two variables have a linear relation is reasonable, for instance by visual inspection. For combinations of common variables for which a non-linear relation is more suitable, other, more advanced modelling approaches could be applied in a similar way. For instance, decision tree models naturally lead to restrictions of the form (7). It may also be useful to fit separate models for subpopulations based on NACE code and/or size class. In general, machine learning techniques could be useful for finding interesting new edit rules (Dumpert, 2020).

B. Choosing reliability weights

32. For reliability weights, we face a similar issue as for edit rules: much already tends to be known about the relative reliability of observed variables $x_{ij}^{(p)}$ within a single data source p, but less is known about the relative reliability of the common variables $y_{il}^{(p)}$ and $y_{il}^{(q)}$ as observed in different data sources. So far, subject-matter experts have provided an initial set of reliability weights $w_l^{(p)}$ for the common variables. Most of these weights are within the range [1, 10], with the exception of two variables from the Statistics on Employees and Salaries which are considered very reliable and were given a weight of 100. However, these weights are

considered to be a highly simplified summary of the quality of each common variable. In reality, the reliability of these variables is expected to vary across different subpopulations of units. A single set of weights cannot account for this. However, concrete information about this variation in reliability is lacking. Again, we focus on data-driven ways to construct better reliability weights.

33. Liepins (1980) showed that a solution to the Fellegi-Holt-based error localization 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 for unit i are chosen as $w_{ij} = -\log\left(\frac{p_{ij}}{1-p_{ij}}\right)$, where p_{ij} denotes the probability that an error has occurred in x_{ij}. Thus, one way to obtain more informative reliability weights would be to take a data set in which the errors are known and model the error probabilities p_{ij} 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.

34. If modelling the error probabilities is not feasible, other approaches could also be used to construct reliability weights for $y_{il}^{(p)}$. Here, a distinction is made between approaches that construct static reliability weights per stratum and approaches that construct dynamic reliability weights that can vary per unit.

35. An indirect way to construct static reliability weights could work by first modelling the occurrence of (large) differences between values $y_{il}^{(p)}$ and $y_{il}^{(q)}$ of the same common variable in different sources, as a function of known background variables such as NACE code, size class, structural complexity of a unit, foreign ownership, etc. For instance, logistic regression, decision trees or more advanced machine learning techniques could be used to identify subpopulations of units for which it is likely that $y_{il}^{(p)} \gg y_{il}^{(q)}$ and other subpopulations where it is likely that $y_{il}^{(p)} \ll y_{il}^{(q)}$. The results of this analysis are then discussed with subject-matter experts to (hopefully) decide which of the observed values is more likely to be correct in these scenarios. These scenarios are then used to define criteria for adjusting the reliability weights per subpopulation. (In addition, these discussions with subject-matter experts could also yield new deductive correction rules or edit operations for the extended error localization problem.)

36. A data-driven way to construct dynamic reliability weights for $y_{il}^{(p)}$ could work as follows:

1) Split the data set containing $\left(\mathbf{y}_i^{(1)}, \ldots, \mathbf{y}_i^{(P)}\right)$ into records that satisfy all edit rules (6*) and (7) and records that violate at least one edit. The first set – which does not require editing during step 1 – is used as reference data for the second set.
2) For each unit i that requires editing, find its nearest neighbour m in the reference data set according to the distance function $d(i,m) = \sum_p \sum_l \left|\tilde{y}_{il}^{(p)} - \tilde{y}_{ml}^{(p)}\right|$, where $\tilde{y}_{il}^{(p)}$ and $\tilde{y}_{ml}^{(p)}$ are standardized versions of $y_{il}^{(p)}$ and $y_{il}^{(p)}$ so all variables are a priori equally important for the distance function. (Here, a robust form of standardization could be applied by subtracting the median of each variable and dividing by the interquartile range.) In addition, we may try to find the nearest neighbour within the same stratum as i by NACE code and/or size class, provided sufficient reference units are available.
3) Let $r_{il}^{(p)} = \left|\tilde{y}_{il}^{(p)} - \tilde{y}_{ml}^{(p)}\right| / d(i,m)$ so that $\sum_p \sum_l r_{il}^{(p)} = 1$. Define the reliability weights as a monotonically decreasing function of $r_{il}^{(p)}$. That is to say, a variable is considered less reliable if it contributes more to the total distance of record i to its nearest neighbour.

A similar approach to construct dynamic reliability weights was tested previously in a single-source editing context for Structural Business Statistics, where it worked reasonably well (Scholtus, 2010).

37. For both approaches, a relevant question is how much the initial reliability weights $w_l^{(p)}$ should be adjusted based on stratum- or 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. For static weights as proposed above, one approach could be to reduce (or increase) a reliability weight by a certain fixed factor for each criterion that is satisfied. Another approach could be to define a limited set of possible values for reliability weights and shift a weight to a value with a lower (or higher) rank for each

criterion that is satisfied. Similarly, dynamic reliability weights could also be restricted to a limited set of possible values.

IV. Evaluating the quality of automatic multisource editing

38. A natural way to evaluate the quality of automatic editing is by comparing automatic editing to manual editing, under the assumption that the 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 of the whole population. Second, due to the isolated nature of current production processes (as discussed in Section I), manual editing on historical data was often done without taking consistency across different statistics explicitly into account. Therefore, these data may not be considered as a 'gold standard' for our purposes.

39. 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 seven different economic sectors. 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 units that are part of a larger enterprise group were also excluded.

40. It remains to be seen for how many sampled units 'gold standard' data can be obtained through this manual editing exercise. For economic sectors where a sufficiently large subsample of 'gold standard' data is 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 accuracy. Another interesting measure is the percentage of records for which exactly the right error pattern was found (Daalmans and Scholtus, 2018). The distributions of values after automatic and manual editing can also be compared 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).

41. In addition to these direct evaluations, a more indirect way to evaluate the effects of automatic editing is to compare aggregated statistics such as stratum totals before and after editing. Any large changes in these statistics due to automatic editing are acceptable only if these are considered plausible by subject-matter experts. With this in mind, the subpopulations found by the analysis discussed in paragraph 35 are also relevant here, because we can check whether automatic editing has indeed adjusted the data for these subpopulations according to the expectations of the subject-matter experts. Visualizations of the data before and after editing can also be useful, to highlight unexpected patterns in the adjustments made by automatic editing.

V. References

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

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.
N. Mushkudiani, J. Daalmans and J. Pannekoek (2014), Macro-Integration for Solving Large Data Reconciliation Problems. *Austrian Journal of Statistics* **43**, 29–48.
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 (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).
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).
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.