Automated and manual data editing: a view on process design and methodology

Jeroen Pannekoek, Sander Scholtus and Mark van der Loo

Discussion paper (201309)

Statistics Netherlands

The Hague/Heerlen, 2013

Explanation of symbols

.	data not available
*	provisional figure
**	revised provisional figure (but not definite)
x	publication prohibited (confidential figure)
–	nil
–	(between two figures) inclusive
0 (0.0)	less than half of unit concerned
empty cell	not applicable
2012–2013	2012 to 2013 inclusive
2012/2013	average for 2012 up to and including 2013
2012/'13	crop year, financial year, school year etc. beginning in 2012 and ending in 2013
2010/'11– 2012/'13	crop year, financial year, etc. 2010/'11 to 2012/'13 inclusive

Due to rounding, some totals may not correspond with the sum of the separate figures.

Publisher
Statistics Netherlands
Henri Faasdreef 312
2492 JP The Hague

Prepress
Statistics Netherlands
Grafimedia

Cover
Teldesign, Rotterdam

Information
Telephone +31 88 570 70 70
Telefax +31 70 337 59 94
Via contact form:
www.cbs.nl/information

Where to order
E-mail: verkoop@cbs.nl
Telefax +31 45 570 62 68

Internet
www.cbs.nl

ISSN: 1572-0314

60083201309 X-10

Automated and manual data editing: a view on process design and methodology

Jeroen Pannekoek, Sander Scholtus, and Mark van der Loo

Data editing is arguably one of the most resource-intensive processes at NSIs. Forced by ever increasing budget pressure, NSIs keep searching for more efficient forms of data editing. Efficiency gains can be obtained by selective editing, that is limiting the manual editing to influential errors, and by automating the editing process as much as possible. In our view, an optimal mix of these two strategies should be aimed for. In this paper we present a decomposition of the overall editing process in a number of different tasks and give an up-to-date overview of all the possibilities of automatic editing in terms of these tasks. In designing an editing process, this decomposition enables one to decide which tasks can be done automatically and for which tasks (additional) manual editing is required. Such decisions can be made a priori, based on the specific nature of the task, or by empirical evaluation, which is illustrated by examples. The decomposition in tasks, or statistical functions, also naturally leads to reuseable components, resulting in efficiency gains in process design.

This paper has been submitted for publication in a special issue of the Journal of Official Statistics on selective editing.

Keywords: automatic editing; selective editing; edit rules; process design; process evaluation

Contents

1 Introduction

The quality of raw data available to National Statistical Institutes (NSIs) is rarely sufficient to allow of the immediate production of reliable statistics. As a consequence, NSIs often spend considerable effort to improve the quality of micro-data before further processing can take place.

Statistical data editing encompasses all activities related to the detection and correction of inconsistencies in micro-data, including the imputation of missing values. Data editing, or at least the correction part of data editing, has traditionally been performed manually by data editing staff with subject-specific expert knowledge. The manual follow-up of a large number of detected inconsistencies is, however, very time-consuming and therefore expensive and it decreases the timeliness of publications. Therefore, several approaches have been developed to limit this very resource-consuming manual editing.

One approach is selective editing (Latouche and Berthelot, 1992). This is an editing strategy in which manual editing is limited or prioritised to those errors where this editing has a substantial effect on estimates of the principal parameters of interest. Provided that there is an effective way of determining the influential errors, this strategy can be successful because it has been well-established (see the review by Granquist and Kovar (1997)) that for many economic surveys only a minority of the records contains influential errors that need to be edited; the remaining errors can be left in without substantial effect on the principal outputs.

An alternative route to reducing manual editing is to perform the editing automatically. Automatic editing is not a single method but consists of a collection of formalised actions that each perform a specific task in the overall editing process. Some well-known tasks that are performed in automatic editing are the evaluation of edit rules to detect inconsistencies in the data, the localisation of fields that cause these inconsistencies, the detection and correction of systematic errors such as the well-known thousand error, and the imputation of missing or incorrect values. Once implemented, automatic editing is fast, uses hardly any manual intervention and is reproducible. For reasons of efficiency, it should therefore be preferred to manual editing even if the latter is confined to selected records. However, not all data editing functions can be performed automatically with sufficient quality of the result. Selective manual editing is then a necessary addition.

Both selective editing and rule-based automated data editing are well-established techniques that have been in use for several decades now. Forced by ever decreasing budgets as well as the pressure to minimise administrative burden, NSIs need to keep searching for more efficient ways to produce statistics, including

more efficient forms of data editing.

The relation between manual and automatic editing as it emerges from the classical literature on selective editing is that all important amendments should be done manually and that the role of automatic editing is confined to the less influential errors: its purpose is mainly to ensure internal consistency of the records so as to avoid inconsistencies at all levels of aggregation. In this view the quality of automatic editing has no bearing on the decision to edit a record manually or automatically. Efficiency gains are realised by the selection process only. The point of view taken in this paper is that for reasons of efficiency, manual editing should be confined to the data that are influential *and* cannot be treated automatically with sufficient quality. In this view, the quality of automatic editing is important in making the decision to edit manually or not and improvements in automatic editing will lead to efficiency gains.

This paper gives an overview of the current state of the art in efficient editing of establishment data. Using numerical results from two example statistics, it is shown that with the current methods, selective editing can be minimised while data quality is retained. We identify methodological research directions which in our view have potential for yielding further efficiency gains.

Besides making the data editing process more efficient, there is a need for increasing the cost-effectiveness of designing and implementing data editing systems. In this paper we propose a hierarchical decomposition of the data editing process into six different task types, called *statistical functions*. This view of the overall process builds on previous work of Camstra and Renssen (2011) and Pannekoek and Zhang (2012) by adding a taxonomy of editing functions and defining the minimal input and output requirements of each of these functions. Identifying the in- and output parameters of these abstract functions allows one to move towards a modern approach to process design, based on reusable components that connect in a plug-and-play manner.

The remainder of this paper is structured as follows. Section 2 discusses some basic aspects of error detection in manual and automatic editing. First we consider the different kinds of errors that can arise and differentiate between errors for which automatic treatment is a possibility and those for which manual treatment is required. Then we discuss the edit rules that are extensively used in data editing, in particular with respect to business surveys. In section 3 an overview is given of both well-known and more recently developed automatic error detection and correction methods. Section 4 is concerned with a decomposition of the overall data editing process in data editing functions based on the action and purpose of these functions. In section 5 the application of a sequence of different editing functions is illustrated using two real data examples. This section also gives references to the freely available R packages that are used for

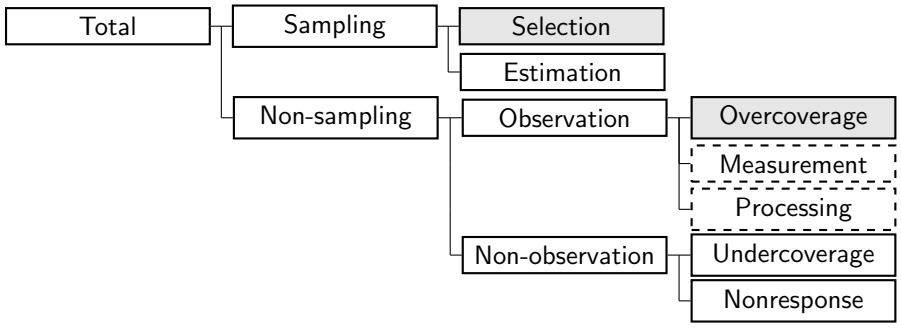

Figure 1. Bethlehem (2009) taxonomy of survey errors. Errors in grey boxes are commonly solved by manual data editing while automated techniques are usually more suited for error causes indicated in dotted boxes.

these illustrations. Finally, in section 6 we summarise some conclusions.

2 Error detection in manual and automated editing

2.1 Sources of errors in survey data

In analyses of survey errors it is customary to decompose the total error into more or less independent components which may be treated or solved separately. Well-known decompositions include those by Groves (1989) and Bethlehem (2009). Here, we use Bethlehem's taxonomy of survey errors since it allows us to identify sources of error with common data editing strategies.

Bethlehem (2009) uses the scheme shown in Figure 1 to distinguish between sources of error in a statistical statement based on surveys. The total error is decomposed into sampling and nonsampling error. The sampling error is further decomposed into selection and estimation error. Selection error consists of differences between the theoretical and realised inclusion probabilities of sampling units, while estimation error consists of the usual variance and bias introduced by the estimation method. Non-sampling errors can be split into observational and non-observational errors. Observational errors are composed of overrepresentation of elements in the population register (overcoverage), measurement errors (item non-response, completion errors, *etc.*) and processing errors at the NSI (*e.g.* data entry errors). Non-observational errors are caused by omission of elements from the population register (undercoverage) and unit non-response.

Traditionally, automated data editing methods have more or less focused on errors happening at the measurement or processing stage. That is, many automated data editing methods focus on the observed variables rather than the identifying or classifying variables already available in the population register. For example, in a stratified hot-deck imputation scheme, the values of stratify-

ing variables are assumed correct to begin with. In contrast, data editing staff often do not make such assumptions and may frequently reclassify units.

Since automated data editing methods are always based on mathematical modeling they usually assume that some kind of structured auxiliary information is available. In many cases historic records, auxiliary register variables or totals from related statistics can be used to estimate values for erroneous or missing fields in a survey data set. In contrast, data editing staff may use unstructured auxiliary information to edit records. Such information may, for example, include written financial reports or information from websites, as well as recontacts. These two differences between manual and automated data editing enable data editing staff to correct for errors not caused at the moment of measurement.

In 2010, thirteen of Statistics Netherlands' data editing employees working on the short term business survey were informally interviewed on commonly found errors and data editing practices. Besides a number of commonly found measurement errors (reporting of net instead of gross turnover, reporting of value of goods instead of invoices, *etc.*) many causes of error that were mentioned are non-observational or sampling errors in Bethlehem's taxonomy. Examples include misclassifications such as retailers being registered as wholesalers, population effects such as bankruptcies, splits and mergers, and differences between legal units (chambre of commerce), tax units (of the tax office) and economic units (of Statistics Netherlands). Such errors are detected and/or solved by looking at auxiliary information such as figures and articles from sector organisations and (financial) newspapers, a website dedicated to registering bankruptcies, publicly available information on wages and retirement funds in a sector and so on. Subject-matter experts also use (often unstructured) domain knowledge on branche-specific transient or seasonal effects to detect errors. Examples of such effects include weather conditions (energy and construction), holidays (food industry, printers, *etc.*) and special events (tourist sector).

For the various measurement errors mentioned by the interviewees, conventional automatic data editing methods can in principle be applied. For non-observational errors like population errors and misclassifications, the error detection and correction process is based on fuzzier types of information and therefore harder to automate. At the moment, we are not aware of methods that can exploit such information for data editing purposes automatically.

2.2 Edit rules for automatic verification

Prior knowledge on the values of single variables and combinations of variables can be formulated as a set of edit rules (or edits for short), which specify or constrain the admissible values. For single variables such edits are range checks;

for most variables in business surveys these amount to a simple non-negativity requirement such as:

e_1 : *Number of employees* ≥ 0
e_2 : *Turnover* > 0

Edits involving multiple variables describe the admissible combinations of values of these variables in addition to their separate range restrictions. For numeric business data, many of these edits take the form of linear equalities (balance edits) and inequalities. Some simplified examples of such edit rules are:

e_3 : *Result* = *Total revenues* − *Total costs*
e_4 : *Total costs* = *Purchasing costs* + *Personnel costs* + *Other costs*
e_5 : *Turnover* = *Turnover main activity* + *Turnover other activities*
e_6 : *Employee costs* $< 100 \times$ *Number of employees*

The inequality and equality edits e_1–e_5 are examples of *fatal* or *hard* edits: they must hold true for a correct record. This class of edits is opposed to the so-called *soft* or *query* edits whose violation points to highly unlikely or anomalous (combinations of) values that are suspect to be in error although this is not a logical necessity. The edit e_6 could be interpreted as a soft edit.

More generally, an inequality edit k can be expressed as $\sum_{j=1}^{J} a_{kj}x_j \leq b_k$, with the x_j denoting the variables, the a_{kj} coefficients, b_k a constant and the summation running over all variables. In e_1 and e_2, $b_k = 0$ and the a_{kj} are zero for all variables except one, for which a_{kj} is -1. Linear equalities such as e_3, e_4 and e_5 can similarly be expressed as $\sum_{j=1}^{J} a_{kj}x_j = b_k$.

Notice that these edits are connected by certain common variables, which is true for many of the edits used in business statistics and has consequences for error localisation and adjustment for consistency. In such situations it is convenient to re-express the edits as a system of K linear equations and inequalities, in matrix notation:

$$\boldsymbol{E}\boldsymbol{x} \odot \boldsymbol{b}, \tag{1}$$

with $\boldsymbol{E}$ the $K \times J$ *edit matrix* with elements a_{kj}, $\boldsymbol{x}$ a J-vector containing the variables and $\boldsymbol{b}$ a K-vector with elements b_k. The symbol $\odot$ should here be interpreted as a vector of operators (with values $<$, $=$ or $\leq$) appropriate for the corresponding (in)equalities.

Each of the edit rules can be verified for each record. If we have N records and K edits, all the failure statuses can be summarised in a binary $N \times K$ *failed-edits matrix* $\boldsymbol{F}$, corresponding to all the record-by-edit combinations. The failure statuses can be the input to an error localisation function that selects variables, from those involved in failed edits, with values that are to be considered

erroneous and need to be changed in order to resolve the edit failures (see Section 3.3).

The number of edit rules greatly varies between statistical domains. The structural business statistics (SBS) are an example with a large number of edit rules. An SBS questionnaire can be divided into sections. It contains, for instance, sections on employees, revenues, costs and results. In each of these sections a total is broken down in a number of components. Components of the total number of employees can be part-time and full-time employees and components of total revenues may be subdivided in turnover and other operating revenues. The total costs can have as components: purchasing costs, depreciations, personnel costs and other costs. The personnel costs can be seen as a subtotal since it can again be broken down in subcomponents: wages, training and other personnel costs. Each of these breakdowns of a (sub)total corresponds to a (nested) balance edit. SBS questionnaires also contain a profit and loss section where the revenues are balanced against the costs to obtain the results (profit or loss), which leads to the edit e_3. This last edit connects the edits from the costs section with the edits from the revenues section. Soft edits for the SBS form are often specified as bounds on ratios. For instance, ratios between a component and the associated total, between the number of employees and the personnel costs, between purchasing costs and turnover, *etc.*

3 Methods for automatic detection and amendment of missing or erroneous values

The overall editing process can be seen as a sequence of statistical functions applied to a data set. Such functions, for example selecting records for manual editing, may be implemented as an automated or a manual subprocess. In this section we summarise a number of data editing methods that can be performed automatically.

Since these methods often detect or correct different types of errors, they will usually be applied one after another so as to catch as many errors as possible. The detailed exposition of the statistical methodology for each of these functions is beyond our scope but below we summarise the type of methods that could be used and/or give some simple examples. More detailed descriptions can be found in De Waal et al. (2011) and the references cited there.

3.1 Correction of generic systematic errors

From a pragmatic point of view, a systematic error is an error for which a plausible cause can be detected and knowledge of the underlying error mechanism

enables a satisfactory treatment in an unambiguous deterministic way. De Waal et al. (2012) distinguish between generic systematic errors and subject-related systematic errors. A generic systematic error is an error that occurs with essentially the same cause for a variety of variables in a variety of surveys or registers. Subject-related systematic errors on the other hand occur for specific variables, often in specific surveys or registers.

3.1.1 Unit of measurement error

A well-known generic systematic error is the so-called unit of measurement error which is the error of, for example, reporting financial amounts in Euros instead of the requested thousands of Euros. Unit of measurement errors are often detected by a simple ratio criterion that compares the raw value x_{raw} with a reference value x_{ref}. Such a rule can be expressed as

$$\frac{x_{raw}}{x_{ref}} > t, \tag{2}$$

with t some threshold value. The reference value can be an approximation to the variable x that is unaffected by a unit of measurement error, such as an edited value for the same unit from a previous round of the same survey or a current or previous stratum median of x. The detection of unit of measurement errors may be improved by dividing the financial variables by the number of employees (*e.g.* Costs or Revenues per employee) to eliminate the variation in these variables due to the size of the unit. If a thousand error is detected, the affected values are divided by thousand. See *e.g.* Di Zio et al. (2005) and Al Hamad et al. (2008) for further discussion and more advanced methods for detecting unit of measurement errors.

Thousand errors are often made in a number of financial variables simultaneously, yielding what is known as a uniform thousand error in these variables. Thousand errors will not violate balance edits if they are uniform in all variables involved; therefore, they cannot be detected by such edits. Incidental thousand errors may be detected by balance edits when the error is made in one or more of the components or their total but not in all these variables.

3.1.2 Simple typing errors, sign errors and rounding errors

Some inconsistencies are caused by simple typing errors. Recently, methods have been developed to reliably detect and correct these types of errors (Scholtus, 2009; Van der Loo et al., 2011). The algorithm correcting for typing errors uses the edit rules to generate candidate solutions and accepts them if the difference with the original value is not larger than a pre-specified value. The difference

is measured with the restricted Damerau-Levenshtein distance (Damerau, 1964; Levenshtein, 1966). This distance measure counts the (possibly weighted) number of deletions, insertions, alterations and transpositions necessary to turn one character string into another (the restriction entails that substrings, once edited, cannot be edited again).

The typo-correction can also correct simple sign errors. More complex sign errors, such as those caused by swapping *Cost* and *Turnover* in a questionnaire where the rule $Profit = Turnover - Cost$ must hold, can be solved by a binary tree algorithm that tests whether (combinations of) swapping options decrease the number of violated edits (Scholtus, 2011).

Rounding errors cause edit violations by amounts of a few units of measurement at most. It is therefore of less importance which variables are adapted. The scapegoat algorithm of Scholtus (2011) uses a randomisation procedure to adapt one or more variables by a small amount such that the number of equality violations is decreased.

3.2 Domain-specific correction rules

In contrast to generic systematic errors, subject-related or domain-specific systematic errors occur for specific variables, often in specific surveys or registers. Problems with understanding definitions are often a cause of such errors. Restaurants, for instance, often incorrectly classify their main revenues as revenues from trade (because they sell food) rather than revenues from services as it should be. As another example, reporting net rather than gross turnover may occur frequently in some domains.

Direct if-then rules can easily be used to correct such errors. These rules are of the form

if *condition* then *action*,

where *condition* is a logical expression that is true if an error is detected and *action* is the amendment function that assigns new values to one or more variables.

Apart from being used for correction of subject-specific systematic errors, such rules are also used for selection and imputation. For selection of records for manual editing, the action consists of assigning TRUE to an indicator variable for manual treatment. For instance, if for large units crucial variables such as Employment or Turnover are missing or inconsistent, the unit may be selected for manual treatment. For the selection of fields to be changed the action consists of changing some fields to NA (which stands for Not Available or missing).

For instance, if the costs per employee are outside the admissible range, the number of employees (in FTE) may be selected as erroneous rather then the employee costs because it is known that the financial variables are reported more accurately. For imputation the condition specifies which missing value can be imputed by the rule and under what conditions. For instance

> if *Wages for temp. employees* = NA and *Nr. of temp. employees* = 0
> then *Wages for temporary employees* $\equiv 0$,

We use the symbol $\equiv$ when we need to distinguish assignment from mathematical or logical equivalence (=). Even the evaluation of an edit rule can be seen as a rule in this if-then form. The condition is in that case the edit rule itself and the action is the assignment of a TRUE–FALSE status to a column of the matrix $\boldsymbol{F}$.

These rules are called *direct* correction/selection/imputation rules because the implementation of the condition and the action follows trivially from the rule itself. In contrast, the generic systematic errors discussed above such as typos and rounding errors are also based on rules, because they use the edit rules, but in those cases the implementation cannot be formulated in a single simple if-then rule but requires a more sophisticated algorithm. The same is true for Fellegi-Holt-based error localisation and model-based imputation with estimated parameters, to be discussed below.

3.3 Error localisation

Error localisation is the process of pointing out the field(s) containing erroneous values in a record. Here, we assume that all fields should be filled, so an empty field (NA) is also assumed erroneous. If there are N records with J variables, the result of an error localisation process can be represented as a boolean $N \times J$ matrix $\boldsymbol{L}$, of which the elements L_{ij} are TRUE where a field is deemed erroneous (or when it is empty) and FALSE otherwise.

Automated error localisation can be implemented using direct rules, as mentioned in section 3.2. In such a case a rule of the form

$$\text{if } \textit{condition} \quad \text{then } L_{ij} \equiv \text{TRUE}, \tag{3}$$

can be applied. It should be noted that this method takes no account of edit restrictions, and does not guarantee that a record can be made to satify all the edits by altering the content of fields pointed out with this method; Boskovitz (2008) calls this the *error correction guarantee*.

Error localisation becomes more involved when one demands that 1) it must be possible to impute fields consistently with the edit rules and 2) the (weighted)

number of fields to alter or impute must be minimised. These demands are referred to as the principle of Fellegi and Holt (1976). Identifying 1 and 0 with the boolean values TRUE and FALSE respectively, the localisation problem for each row $\boldsymbol{l}$ of $\boldsymbol{L}$ can be denoted mathematically as

$$\boldsymbol{l} \equiv \underset{\boldsymbol{u}\in\{0,1\}^J}{\arg\min}\, \boldsymbol{w}^T\boldsymbol{u} \tag{4}$$

under the condition that the set of (in)equality restrictions Eq. (1) has a solution for the x_j with $l_j = 1$, given the original values of the x_j with $l_j = 0$. The vector $\boldsymbol{l}$ points out which variables are deemed wrong (1) and which are considered correct (0). In addition, $\boldsymbol{w}$ is a non-negative weight vector assigning weights to each of the J variables. These weights are referred to as reliability weights, because they can be used to express the degree of trust one has in each original value x_j. Note that increasing w_j makes it less likely that x_j will be chosen as a candidate for amendment, as feasible solutions with lower weights are more likely to be available.

A special case occurs when only univariate (range) edits are considered. That is, when every edit contains but a single variable. Denote by $\boldsymbol{C}$ the $K \times J$ boolean matrix that indicates which variables (columns) occur in which edits (rows), and denote by $\boldsymbol{X}$ the $N \times J$ numerical data matrix. In this special case, the matrix $\boldsymbol{C}$ is either diagonal (when all variables are bounded from above or below), or contains at most $2J$ nonzero elements (when each variable is bounded by a range). The matrix $\boldsymbol{L}$ can then be computed as

$$\boldsymbol{L} \equiv (\boldsymbol{F}\boldsymbol{C} > 0) \vee (\boldsymbol{X} = \mathsf{NA}). \tag{5}$$

Here, $\boldsymbol{F}$ is the $N \times K$ failed-edits matrix defined in Section 2.2, and the logical and comparison operators ($<$ and $=$) on the right-hand-side should be evaluated elementwise. The symbol $\vee$ indicates the elementwise or operation.

Several algorithms have been developed for error localisation under interconnected multivariate linear constraints. See De Waal et al. (2011) and the references therein for a concise overview of available algorithms. Regardless of the algorithm used, the special case of Eq. (5) can be applied to the subset of univariate edits prior to one of the more complex algorithms, to reduce computational complexity. The branch-and-bound algorithm of De Waal and Quere (2003) and approaches based on a reformulation of the error localisation problem as a mixed-integer problem (MIP) have recently been implemented as a package for the R statistical environment by De Jonge and Van der Loo (2011).

3.4 Imputation of missing or discarded values

Imputation is the estimation or derivation of values that are missing due to non-response or discarded for being erroneous (as indicated by $\boldsymbol{L}$ in the previous

section). Below we discuss deductive and model-based imputation methods.

3.4.1 Deductive imputation of missing or discarded values

In some cases the values for the empty fields can be derived uniquely from edit rules by mathematical or logical derivation. For example, when one value in a balance edit is missing, the only possible imputed value that will satisfy the balance edit can easily be obtained from the observed values. For the interrelated systems of linear edits that are typical for the SBS it is generally not obvious if some of the missing values are determined uniquely by the edit rules. By filling in the observed values from a record in the edit rules, a system of (in)equalities is obtained with the missing values as unknowns. Specifically, if $\boldsymbol{x}$ is partitioned as $\boldsymbol{x} = (\boldsymbol{x}_{obs}, \boldsymbol{x}_{mis})$ where $\boldsymbol{x}_{obs}$ denotes the sub-vector of $\boldsymbol{x}$ containing the observed values and $\boldsymbol{x}_{mis}$ the sub-vector with missing values and $\boldsymbol{E}$ is partioned conformably as $\boldsymbol{E} = (\boldsymbol{E}_{obs}, \boldsymbol{E}_{mis})$, then we have from $\boldsymbol{E}\boldsymbol{x} \odot \boldsymbol{b}$,

$$\boldsymbol{E}_{mis}\boldsymbol{x}_{mis} \odot \boldsymbol{b} - \boldsymbol{E}_{obs}\boldsymbol{x}_{obs}, \tag{6}$$

where the right hand side is calculated from the observed values and $\boldsymbol{x}_{mis}$ contains the unknown missing values. The problem now is to determine which, if any, of these unknowns can be solved from this system and consequently deductively imputed. There exist simple algorithms that can find the values of all uniquely determined values for the unknowns in this system (De Waal et al., 2011).

3.4.2 Model-based imputation

Deductive imputation will in general only succeed for part of the missing values. For the remaining missings, models are used to predict the values of the missing items and these predictions are used as imputations. Here the term "model" is used in a broad sense, covering not only parametric statistical models but also non-parametric approaches such as nearest-neighbour imputation.

For business surveys with almost exclusively numerical variables, the predominant methods are based on linear regression models including, as special cases, (stratified) ratio and mean imputation (*cf.* De Waal et al. (2011, ch. 7)). Important for the efficiency of the application of regression imputation is that models for each of the variables that need imputation are specified in advance or selected automatically without the need for time-consuming model selection procedures by analysts at the time of data editing. When available, a historical value is often a good predictor for the current value.

An alternative, if all variables are continuous, is to use a multivariate regression approach where all variables that are observed in a record are used as predictors

for each of the missing values. Thus, for each record, the variables are partitioned in two sets; the variables observed in record i and the variables missing in that record. The subvectors of $\boldsymbol{x}$ corresponding to these two sets will be denoted by $\boldsymbol{x}_{obs(i)}$ and $\boldsymbol{x}_{mis(i)}$ and the value of $\boldsymbol{x}_{obs(i)}$ in record i by $\boldsymbol{x}_{i.obs}$. If it is assumed that $\boldsymbol{x}$ is multivariate normally distributed, the conditional mean of the missing variables, given the values of the observed variables in record i, $\boldsymbol{\mu}_{i.mis}$ say, can be expressed as

$$\boldsymbol{\mu}_{i.mis} = \boldsymbol{\mu}_{mis(i)} + \boldsymbol{B}_{mis(i),obs(i)}(\boldsymbol{x}_{i.obs} - \boldsymbol{\mu}_{obs(i)}), \tag{7}$$

with $\boldsymbol{\mu}_{mis(i)}$ and $\boldsymbol{\mu}_{obs(i)}$ the unconditional means of $\boldsymbol{x}_{mis(i)}$ and $\boldsymbol{x}_{obs(i)}$ and $\boldsymbol{B}_{mis(i),obs(i)}$ an $n_{mis(i)} \times n_{obs(i)}$ matrix with rows containing the coefficients for the $n_{mis(i)}$ regressions of each of the missing variables on the observed ones. Estimates of the conditional means $\boldsymbol{\mu}_{i.mis}$ are the regression imputations and can be applied for continuous variables for which the linear model is a good approximation, without necessarily assuming normality.

An estimator of the coefficient matrix $\boldsymbol{B}_{mis(i),obs(i)}$ can be obtained from an estimator of the covariance matrix $\boldsymbol{\Sigma}$ of $\boldsymbol{x}$ by using

$$\boldsymbol{B}_{mis(i),obs(i)} = \boldsymbol{\Sigma}^{-}_{obs(i).obs(i)}\boldsymbol{\Sigma}_{obs(i).mis(i)} \tag{8}$$

with $\boldsymbol{\Sigma}_{obs(i).obs(i)}$ the submatrix of $\boldsymbol{\Sigma}$ containing the (co)variances of the variables observed in record i and $\boldsymbol{\Sigma}_{obs(i).mis(i)}$ the submatrix containing the covariances among the variables observed in record i and the variables missing in this record. Note that once we have estimated the covariance matrix $\boldsymbol{\Sigma}$ and mean vector $\boldsymbol{\mu}$ for all variables, we can perform all regressions needed to impute each of the records, with their different missing data patterns, by extracting the appropriate submatrices and subvectors. In (8) we used a generalised inverse, denoted by "$^-$", instead of a regular inverse because the covariance matrix involved can be singular due to linear dependencies of the variables implied by equality constraints.

A nice property of this multivariate regression approach with all observed variables as predictors is that linear dependencies in the data used to estimate $\boldsymbol{\Sigma}$ will be transferred to each imputed record. Therefore, all equality edits will be satisfied by the imputed data provided that $\boldsymbol{\Sigma}$ is estimated on data consistent with these edits (*cf.* De Waal et al. (2011, ch. 9)). A possible data set to be used for estimation is the set of complete and consistent records from the current data. If there are not (yet) enough of such records, cleaned data from a previous round of the survey provide an alternative. If the current data are used it is possible to also include the records with missing values in the estimation of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ by applying an EM algorithm (see Little and Rubin (2002)).

3.5 Adjustment of imputed values for consistency

Imputed values will often violate the edit rules since most imputation methods do not take the edit rules into account. The multivariate regression approach (7) takes equalities into account but not inequalities. More involved imputation methods have been developed that can take all edit rules into account (De Waal et al., 2011, ch. 9), but for many unsupervised routine applications such models become too complex. The inconsistency problem can then more easily be solved by the introduction of an adjustment step in which adjustments are made to the imputed values, such that the record satisfies all the edits and the adjustments are as small as possible. This is an optimisation problem: minimise the adjustments under the constraint that all edits are satisfied. When the weighted least squares criterion is chosen to measure the discrepancy between the unadjusted and the adjusted values, this problem can be formalised as

$$\begin{aligned} \boldsymbol{x}_{adj} = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\arg\min} & (\boldsymbol{x}-\boldsymbol{x}_{unadj})^T \boldsymbol{W} (\boldsymbol{x}-\boldsymbol{x}_{unadj}) \\ \text{subject to } & \boldsymbol{E}\boldsymbol{x}_{adj} \odot \boldsymbol{b}, \end{aligned} \tag{9}$$

where it is understood that only the imputed values may be changed; the other elements of $\boldsymbol{x}_{adj}$ remain equal to the corresponding elements of $\boldsymbol{x}_{unadj}$. The matrix $\boldsymbol{W}$ is a positive diagonal matrix with weights that determine the amount of adjustment for each of the variables; adjustments to variables with large weights have more impact on the criterion value and therefore these variables are adjusted less than variables with small weights. For instance, the choice $\boldsymbol{W} = diag(\boldsymbol{x}_{unadj})^{-1}$ leads to minimisation of the squared discrepancies relative to the size of the unadjusted values; see Pannekoek and Zhang (2011) for more details.

3.6 Selection of units for further treatment

Automatic treatment cannot be expected to find and repair all important errors and consequently some form of additional manual treatment will be needed. The selection of units for manual treatment is the essential part of selective editing. The goal of this approach is to identify units for which it can be expected that manual treatment has a significant effect on estimates of totals and other parameters of interest and to limit manual review to those units.

An important tool in this selection process is the score function (Latouche and Berthelot, 1992; Lawrence and McDavitt, 1994; Lawrence and McKenzie, 2000; Hedlin, 2003) that assigns values to records that measure the expected effect of

editing. The record score is usually built up from local scores for a number of important variables. Each local score measures the significance for the variable of concern. Often it can be decomposed into a *risk* component that measures the size or likelihood of a potential error, and an *influence* component that measures the contribution or impact of that value on the estimated target parameter. The local score for variable j in record i can then be expressed as $s_{ij} = F_{ij} \times R_{ij}$ with F_{ij} the influence component and R_{ij} the risk component for variable j in record i. See, *e.g.*, Di Zio (2013) for an example of a local score function with risk and influence components. A record- or unit-level score is a function of local scores, *i.e.* $S_i = f(s_{i1}, \ldots, s_{iJ})$. The measure of risk is commonly based on the deviation of a variable from a reference value, often a historical value or stratum median. Large deviations from the reference value indicate a possible erroneous value and, if it is indeed an error, a large correction.

Since the local score and the record score reflect the occurrence and size of outlying values with respect to the reference values, the score can be seen as a quantitative measure for an aspect of the quality of a record. In this sense it is a verification function (*cf.* Section 4). Its purpose, however, is selection and this can be accomplished by comparing the scores with a predetermined threshold value and selecting the units with score values higher than the threshold for manual editing. Alternatively, the units can be ordered with respect to their score values and manual editing can proceed according to this ordering, until some stopping criterion is met.

In practice we also see other, simpler, selection functions being applied. The following are some examples.

- A function that identifies units that are 'crucial' because they dominate the totals in their branche; selected units will be reviewed manually, whether they contain suspect values or not (selection on influence only).

- A function that selects influential units for which automatic imputation is not considered an accurate treatment because some main variables are missing or obviously incorrect; selected units will be re-contacted.

- A function that selects non-influential units for which automatic imputation is not considered an accurate treatment because some main variables are missing or obviously incorrect; selected units will be treated as unit non-response, for instance by weighting techniques in the estimation phase after editing is completed.

- A function that selects units for which an automatic action has failed. For instance if the error localisation took too much time and the process was

stopped without having obtained a solution. Selected units can be treated as unit non-response or reviewed manually, depending on their influence.

For some recent theoretical developments in the field of selective editing, see Albuéz et al. (2013) and Di Zio (2013).

4 The data editing process

Much of the complexity in the design of a data editing system is caused not by mathematical difficulties relating to the underlying methods, but by combining the implementation of those methods into a working process or supporting system.

A typical data editing process consists of a mixture of domain-specific error correction and localisation actions, a number of automated editing steps, and a possibility for manual intervention on selected records. Each part of such a process has its own input, output, and control parameters that influence how it can be combined with other steps to build up a full process.

To design, compare and evaluate data editing processes it is useful to have a common terminology for the *types of activities* that are instrumental in realising the end result of a data editing process. In line with Camstra and Renssen (2011) we call these types of activities *statistical functions*. In section 4.1 below we propose a decomposition of the overall data editing process in a taxonomy of statistical functions that are characterised by the kind of task they perform and the kind of output they produce. The effects of these statistical functions can be evaluated by inspecting their characteristic output.

A statistical function describes *what* type of action is performed but leaves unspecified *how* it is performed. To implement a statistical function for a specific data editing application (discussed in section 4.2), a method for that function must be specified and configured. It should be noted that the same statistical function can, and often will, be implemented by several methods even within the same application. For instance the statistical function (or type of task or kind of activity) *record selection* can be implemented by both a score function methodology and by graphical macro-editing.

An actual implementation of a data editing process can now be seen as a collection of implementations of statistical functions. The overall process can be structured by dividing it into subprocesses or *process steps*, that each implement one or several (but related) statistical functions executed by specified methods. Process steps are application-specific but the statistical functions that they implement are much more general and are used to categorise the kinds of activities

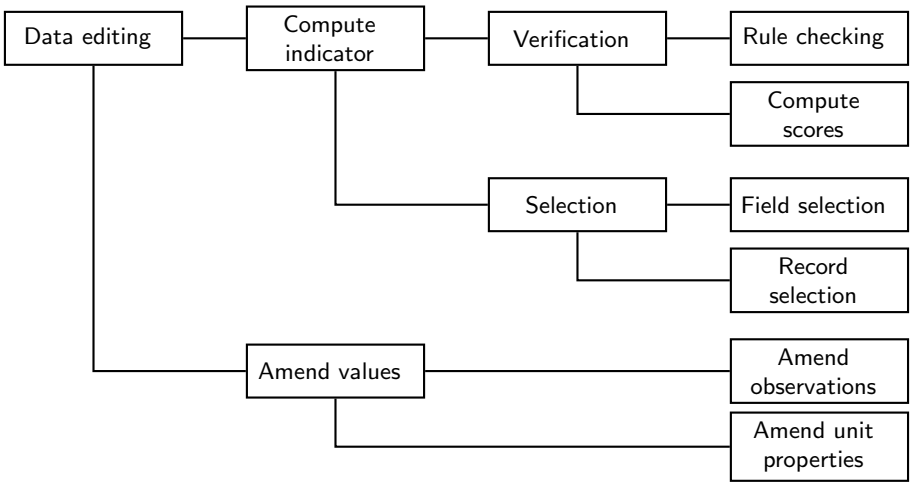

Figure 2. A taxonomy of data editing functions. Each data editing function has its own minimal input-output profile which determine how they may be combined in a data editing process (Table 1).

implemented by the process steps. The granularity in which a process is divided into process steps is, to an extent, arbitrary. For example, one may talk about a statistical process as the complete process from gathering input data to publishing results, and divide that process into process steps using the GSBPM model (UNECE Secretariat, 2009). In that model, data editing occurs as a single step. For our purposes however, it is natural to define a more fine-grained approach.

The choice of methods to be used in the process steps and the order in which the process steps are executed will depend on the properties and requirements of the specific application at hand but some general considerations regarding these choices are discussed in section 4.3.

4.1 A taxonomy of data editing functions

Just like process steps, statistical functions may be separated on several levels of granularity. In Figure 2 we decompose data editing hierarchically, in three levels, into ultimately six low-level statistical functions.

At the first level of the decomposition we distinguish between functions that leave the input data intact (*compute indicator*) and those that alter the input data (*amend values*). At the second level, functions are classified according to their purpose. We distinguish between indicators that are used to verify the data against quality requirements (*verification*) and indicators that are used to separate a record or dataset into subsets (*selection*). *Verification* functions are separated further into functions that verify hard (mandatory) edit rules (*rule checking*) and functions that compute softer quality indicators (*compute scores*). The *selection* function allows for different records (*record selection*) or different

fields in a record (*field selection*) to be treated differently. There is no separation based on purpose for the *amendment* function; *amendment* functions are only separated into functions that alter observed values (*amend observations*) and functions that alter unit properties (*amend unit properties*) such as classifying (auxiliary) variables. This may be interpreted as a decomposition based on a record-wise or field-wise action.

It should be recognised that there are many other dimensions along which one could separate the types of tasks performed in a data editing process. For example, Pannekoek and Zhang (2012) distinguish between methods that can be performed on a per-record basis (*e.g.* Fellegi-Holt error localisation, imputation with historical values) and actions that need batch processing (*e.g.* error localisation by macro-editing, imputation with current means). The point of view we take here is that we wish the taxonomy to abstract from implementation issues. The lowest-level statistical functions defined here allow one to define quality indicators for each function, in terms of their effect on data, performance, expense, *etc.*, which are independent of the chosen statistical method or implementation thereof. Below, the six lowest-level data editing functions are discussed in some detail.

Rule checking. This verification function checks, record by record, whether the value combinations in a record are in the allowed region of the space of possible records. Such a task may be done automatically, when the rules and possible reference data are available in a machine-readable format, or manually, by expert review.

Compute scores. The score function computes a quality indicator of a record or field. Examples of score functions are counting the number of missings in a record, determining whether a field contains an outlier or counting the number of edits violated by a field. The output of score functions is often input for automated selection functions. Score functions are rarely computed manually.

Field selection is used to point out fields in records that need a different treatment than the remaining fields, for example because they are deemed erroneous. Selection may be done manually, by expert review, or automatically. Examples of automated methods include detection of unit of measurement errors, and Fellegi and Holt's method for error localisation.

Record selection aims to select records from a data set that need separate processing. This can be done automatically, for example by comparing the value of a score function to a threshold value. Manual record selection is commonly based on macro-editing methods, such as sorting on a score function, reviewing aggregates, and graphical analyses.

Amend observations. This function aims to improve data quality by altering

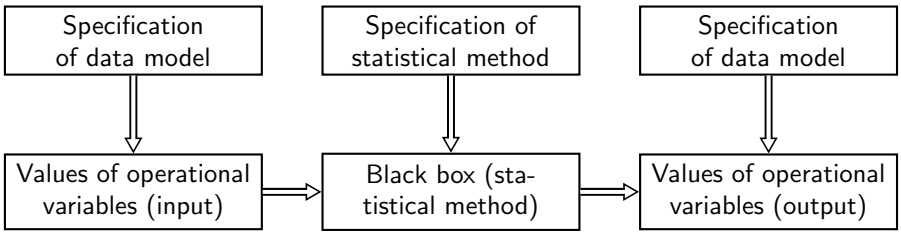

Figure 3. A model to specify the operationalisation of statistical functions (Camstra and Renssen, 2011). Besides statistical data, the values of operational variables *include auxiliary information and control parameters at the input side and process metadata and quality indicators on the output side.*

observed values or by filling in missing values. Many automated imputation and adjustment methods exist, some of which have been discussed in Section 3. The amendment function can also be performed manually, for example by data editing staff who may recontact respondents.

Amend unit properties. This function does not alter the value of observed variables but amends auxiliary properties relating to the observed unit. In business statistics, this function entails tasks like changing erroneous NACE codes and is often performed manually. Another commonly performed task falling into this category is the adjustment of estimation weights for representative outliers.

Pannekoek and Zhang (2012) and Camstra and Renssen (2011) also proposed a decomposition of statistical functions related to data editing. The former distinguish between the *verification*, *selection* and *amendment* functions, while the latter also distinguish *calculation of score functions*. The taxonomy in the current paper further completes the picture by assigning data editing functions a place in a hierarchy based on clearly defined separating principles (amend or not at the first level and select or verify at the second level).

4.2 Specification of data editing functions

As shown above, each function in the taxonomy of Figure 2 can be performed with several methodologies, and each methodology may be implemented in several ways. The operationalisation of a function for a specific data editing process can therefore be specified by documenting the input, output and the method. Indeed, Camstra and Renssen (2011) propose such a specification model for general statistical functions, shown in Figure 3. In principle, a data editing process is completely determined once the order of process steps and their specifications are known.

As an example, consider a simple *record selection* function comparing a score value to a threshold value. The input consists of a score value s and a threshold

value t, so the data model for the input is $\mathbb{R}^2$. The method specification is the algorithm

IF ($s > t$) return(TRUE) ELSE return(FALSE),

so the output data model is {FALSE,TRUE}.

The above algorithm is a very simple example of how a *selection* function may be implemented. In our taxonomy, the work of Di Zio (2013) and Albuéz et al. (2013) presented elsewhere in this issue also falls into the category of *record selection* functions, even though the methods described there are much more advanced. The most important commonality between *record selection* functions is the type of output they produce, namely a decision for each record whether it should be selected or not. Regardless of the method used, such an output can be represented as a boolean vector with the number of records in the data set as dimension. On the input side, any effective *record selection* function will at least need the data to be able to return a reasonable decision vector. At this level of abstraction, even wildly different methods may be compared to support decisions about which method to use in which process step. Indeed, the taxonomy described in this paper has been designed with such a purpose in mind.

Just like for the *record selection* function, it is possible to identify a minimal set of input and output parameters for each data editing function, regardless of the method that implements it. Table 1 denotes this set of minimal in- and output parameters for every low-level statistical function of the taxonomy. Any extra in- or output parameter used in a particular process will be related to the specific method chosen to implement a function. The taxonomy and input-output model presented above make no assumptions about the type of data or type of rule sets. For example, the model leaves undecided whether each data record has the same number of variables, or whether the data have a hierarchical structure (such as used in household surveys). Also, there are no assumptions about the type of rules used; they may be numerical, linear, nonlinear, categorical, of mixed type or otherwise specified.

4.3 Combining process steps

An overall editing process can be seen as a combination of process steps each consisting of one or more statistical functions executed by specified methods. The choice of methods that implement these functions as well as the specifications of parameters or models for these methods will differ between applications and depend on the data to be edited, availability of auxiliary data, output and

Table 1. The minimal input and output for data editing functions. The input data consist of N items, where the number of data attributes may vary per item. Each data attribute is subject to K rules.

Function	**input**	**output**
Rule checking	data, rules	$N \times K$ edit failure indicator
Compute scores	data	N-vector of score values
Field selection	data, rules	field selection indicator
Record selection	data	N-vector of subset indicators
Amend observations	data	data
Amend unit properties	unit properties	unit properties

timeliness requirements, *etc.* Moreover, the order in which process steps will be carried out is also application dependent. However, some general considerations about the composition of process steps in terms of statistical functions, the order of application and the choice of methods will be outlined below.

A single process step can combine several functions that will always be applied together. For instance, correction of generic and domain-specific systematic errors typically involve the implementation of a *field selection* function by a method that detects a specific systematic error and an *amend observations* function to replace the erroneous value with a corrected one. Since the detection is always followed directly by the correction action, specific for the kind of error detected, these two functions are combined in a single process step with data and rules as input and a field selection indicator as well as modified data as output. The indicator reflects the detection part and the modified data the amendment part.

Several process steps will often perform the same statistical function but with different methods. In particular, *amend observations* refers to a large group of process steps that each implement a different method to solve a different problem in (possibly) different data values. An overall process will often include steps that perform the following amendment tasks: correction of generic and domain-specific systematic errors, deductive imputation, model-based imputation and adjustment of imputed values for consistency.

Although the ordering of process steps can differ between applications, there is a logical ordering for some process steps. For instance, selection for interactive treatment itself can occur at different stages of the editing process, but it is evident that for efficiency reasons such a selection step should always precede the actual manual amendment of values and, if the selection is performed by

a score-function methodology, the calculation of scores must precede the selection step. Also, automatic amendment steps will usually start by exhausting the possibilities for solving systematic errors and deductive imputation before approximate solutions by model-based imputation are applied.

Timeliness of the results is an important requirement that influences the choice of methods for the statistical functions. For surveys where the data collection phase extends over a considerable period of time, it is important that the time-consuming manual editing starts as soon as possible, that is as soon as the data are arriving. Selection for manual editing should then be based on a score function that can be evaluated on a record-by-record basis without the need to wait until all or a large part of the data are available. On the other hand, for administrative data or surveys with a short data collection period, selection for interactive treatment can be done using macro-editing methods that by definition use a large part of the data.

5 Numerical illustrations

5.1 Introduction

In this section, we illustrate the effects of applying a sequence of automatic and manual editing functions using two real data sets. Both data sets come from regular production processes at Statistics Netherlands. The first example concerns data on Dutch child care institutions (Section 5.2); the second example concerns SBS data on Dutch wholesalers (Section 5.3).

For both examples, we have identified the following possible process steps that can be applied during editing.

1. Correction of generic and domain-specific systematic errors:
   (a) Correction rules for falsely negative values
   (b) Correction of uniform thousand errors
   (c) Other direct correction rules
   (d) Correction of simple typing errors
   (e) Correction of sign errors
   (f) Correction of rounding errors
2. Automatic error localisation (under the Fellegi-Holt paradigm)
3. Deductive imputation of missing or discarded values
4. Model-based imputation of missing or discarded values

5. Adjustment of imputed values for consistency
6. Selection for interactive treatment
7. Manual editing (interactive treatment)

The first six numbered steps were treated in Section 3 as part of our overview of automatic editing methods. Step 7 is the only one considered here that requires real-time human input. The other steps can be run automatically once they have been set up. As was suggested in Section 4.3, a large number of different editing processes can be obtained by combining some (not necessarily all) of the above process steps, possibly in a different order. In general, different choices will have a different impact on the quality of the output data and on the efficiency of the editing process. This will be illustrated in the examples below.

Some brief remarks on the implementation now follow. All the numerical experiments reported below have been performed in the R statistical environment. Definition and checking of edit rules can be done with the editrules package of De Jonge and Van der Loo (2012). Typing, sign, and rounding errors can be corrected, while taking edit rules into account, with the deducorrect package of Van der Loo et al. (2011). The deducorrect package also offers functionality to reproducibly apply user-defined domain-specific actions, as discussed in Section 3.2. The term "reproducibly" here means that every action performed on the records is automatically logged, while the user can configure the conditional actions independent from the source code defining the data editing process. Error localisation for numeric, categorical or mixed data can be done with the editrules package. See De Jonge and Van der Loo (2011) for an introduction. Deductive imputation methods are again included in the deducorrect package. See Van der Loo and De Jonge (2011) for a description. For model-based imputation a multivariate regression method is applied, implemented in R. Imputed values are adapted using the rspa package of Van der Loo (2012). The code used for selecting records for manual editing and for repairing thousand errors is not part of any package and has been developed for the purpose of this paper.

5.2 Data on child care institutions

In this illustration we will show the effects of a sequence of automatic editing functions in terms of the amount of errors detected and the number of resulting amendments to data values. The data used for this example are taken from a census among institutions for child day care in 2008. Apart from questions on specific activities, the questions and the structure of the questionnaire are similar to what is typical for structural business statistics. For this illustration a subset of the census data was used, consisting of 840 records with 45 variables.

For these variables 40 hard edit rules were specified, of which 11 are equalities, 27 are non-negativity edits and the remaining two are other inequalities. The edit rules as well as the rules for detecting thousand errors and domain-specific generic errors are subsets of the rules used in production.

To these data we have applied the automatic process steps 1 through 5 listed in Section 5.1. The results are displayed in Table 2. The second column of this table shows the number of changed data values at each process step. In the third column are the numbers of failed edits at each process step, which can be obtained directly from the failed-edits matrix. Some edits cannot be evaluated for some records because the edit contains variables with missing values in that record. The corresponding elements of the failed-edits matrix are then missing and the number of such missing elements is in the column *Not evaluated edits*. The number of missing data values is in the last column.

Table 2. Numbers of values changed, edit violations and missings at each step of a sequence of automatic editing functions

Process step	*Changed values*	*Violated edits*	*Not eval. edits*	*Missings*
0. None	0	258	158	124
1a. Rules for false minus signs	9	249	158	124
1b. Thousand errors	17	250	158	124
1c. Other direct rules	43	252	158	124
1d. Simple typing errors	53	187	158	124
1e. Sign errors	0	187	158	124
1f. Rounding errors	102	147	158	124
2. Error localisation	215	0	477	339
3. Deductive imputation	161	0	248	178
4. Model-based imputation	178	109	0	0
5. Adjustment of imputed values	144	0	0	0

The first line of Table 2 shows that before automatic editing there are, in the whole data set, 258 edit violations and 158 edits that cannot be evaluted because of 124 missing values. As a first automatic step, 9 false minus signs are removed by a simple direct rule for a variable that is not part of any equality edit. Obviously 9 non-negativity edit failures are resolved by this action. The detection of uniform thousand errors is applied within the revenues, costs and results section seperately and 17 such errors are found. However, the number of violated edits is increased by one. By looking at the difference between the failed-edits matrix before and after the correction for thousand errors, it appears that the newly failed edit is *Total revenues* − *Total costs* = *Pre-tax result* and that this occurs because a thousand error was detected in the revenues and pre-tax result, but

not in the costs. Records with thousand error corrections that break edit rules should be followed up manually because falsely correcting a thousand error is bound to have influential effects on estimates. The next step concerns the application of other direct rules which results in 43 corrections. Again, some of these changes cause edit failures that should be followed up manually, not only to correct the data but also the see how these direct correction rules can be modified so that they are consistent with the edit rules.

We now apply the algorithms for resolving simple typing errors, sign errors and rounding errors discussed in Section 3.1.2. There are 53 typing errors detected and corrected of which 12 appear to be sign errors. These corrections are very effective in removing errors as the number of violated edit rules is reduced by 65. After the correction of sign errors in step 1a and 1d, the algorithm for more complex sign errors (step 1e) could not detect any additional sign errors. Rounding errors (step 1f) are also important since 40 of the edit violations can be explained by such errors and correcting them with the algorithm mentioned in Section 3.1.2 prevents that these violations need to be treated by the computationally intensive error localisation in step 2. Separating the trivial rounding errors from other, more important, errors also clarifies our picture of the data quality.

At this stage the possibilities for correction of generic and domain-specific systematic errors are exhausted. The remaining inconsistencies and missing values are resolved by applying steps 2 through 5. Error localisation (step 2) identifies 215 values that need to be changed in order to be consistent with all edit rules. These values are treated as missing in the following process steps. The increase of missing values also increases the number of not evaluated edits to a great extent. To impute the missing values, deductive imputation (step 3) is tried first and succeeds in filling in close to half of the missing values with the unique values allowed by the edit rules. For the remaining 178 missing values the multivariate regression method of Section 3.4.2 (step 4) is applied. These imputed values result again in edit violations. However, contrary to the situation prior to step 2, the violation of an edit rule is now not caused by a measurement error in some, probably only a few, of the variables but by the fact that all model-based imputations are only approximations to the real values. Therefore (step 5) we adjust the imputed values as little as possible and solve the 109 edit violations and a complete and consistent data set results.

5.3 Data on Wholesale

For a second illustration, we consider a data set of 323 records from the Dutch SBS of 2007. The data are on businesses with 10 employed persons or more

from the sector wholesale in agricultural products and livestock. The survey contains 93 numerical variables. These should conform to 120 linear edits, of which 19 are equalities.

In terms of the possible process steps listed in Section 5.1, the editing process that was actually used in production consisted of steps 1(abc) and 6, followed by step 7 for the selected records and by steps 2, 4, and 5 for the rest. Selection for interactive treatment was based on a score function for businesses with less than 100 employed persons. Businesses with 100 employed persons or more were always edited manually. In addition, the model-based imputations in step 4 were obtained from a linear regression model with one predictor separately for each variable. We use the outcome of this production process as a benchmark.

The second column of Table 3 shows the mean values of twelve key variables in the production-edited data set. The third column shows the corresponding means for the unedited data (ignoring all missing values). Prior to editing, the means of all financial variables are much too high, which reflects the presence of thousand errors in the unedited data. Moreover, while the production-edited means satisfy basic accounting rules such as *Total operating revenues* = *Net turnover* + *Other operating revenues* (apart from rounding effects), the unedited means do not.

The above editing process involves a substantial amount of manual editing: the number of records selected for interactive treatment was 142, or 44% of all records (representing about 84% of total net turnover in the production-edited data set). We now look at two different set-ups that involve less manual editing. The first alternative editing process is almost entirely automated. It consists of the above numbered process steps 1(abcdef) and 2 through 5 (in that order). Step 7 is included as a fall-back to treat records for which automatic error localisation fails. The second alternative process is almost the same, but we add steps 6 and 7 at the end, with a simple selection mechanism that sends all businesses with 100 employed persons or more to manual editing. Note that both alternative editing processes contain the deductive correction methods 1(def) and a deductive imputation step, which were not used in production. These additional steps are expected to improve the quality of automatic editing.

To compare the outcome of these alternative editing processes to our benchmark, we simulated the results in R. In the implementation of the process steps, we mostly followed the methodology that was originally used in production. We only made changes to the model-based imputation and adjustment steps. For model-based imputation, we did not use a separate regression model for each variable but simultaneous regression with all variables as explained in Section 3.4.2. For the adjustment step, linear optimisation was used in production, but here we used quadratic optimisation as implemented in the rspa package.

Table 3. Unweighted means of (non-missing) values of key variables in the SBS wholesale data. The first ten rows contain rounded multiples of 1000 *Euro; the last two rows are in units.*

Variable	*Benchmark*	*Unedited*		*Alternative I*		*Alternative II*	
Total operating revenues	59342	80151	(+35%)	62180	(+4.8%)	59338	(−0.0%)
Net turnover	59158	79546	(+34%)	61652	(+4.2%)	59157	(−0.0%)
Other operating revenues	185	655	(+255%)	528	(+186%)	182	(−1.7%)
Total operating costs	57795	77996	(+35%)	60314	(+4.4%)	57793	(−0.0%)
Purchasing costs	52864	68703	(+30%)	55359	(+4.7%)	52861	(−0.0%)
Depreciations	302	466	(+54%)	303	(+0.2%)	302	(+0.1%)
Personnel costs	2446	5641	(+131%)	2460	(+0.6%)	2446	(−0.0%)
Other operating costs	2183	3258	(+49%)	2192	(+0.4%)	2185	(+0.1%)
Operating results	1547	2574	(+66%)	1866	(+21%)	1545	(−0.1%)
Pre-tax results	1898	2670	(+41%)	1983	(+4.4%)	1919	(+1.1%)
Employed persons (count)	63.9	64.9	(+1.6%)	64.5	(+1.0%)	64.4	(+0.8%)
Employed persons (FTE)	50.8	49.5	(−2.6%)	47.1	(−7.2%)	47.9	(−5.6%)

Manual editing was simulated by copying the production-edited values. The number of manually edited records under the first alternative strategy was 4 (about 1% of all records, also representing about 1% of total net turnover). Under the second alternative strategy, this number was 34 (about 11% of records, but 55% of total net turnover).

The rightmost columns in Table 3 show the means of key variables for both alternative editing strategies. It is seen that the first alternative yields large differences with respect to the benchmark for several variables. Moreover, with one exception all differences are positive. Thus it appears that for this data, relying completely on automatic editing does not produce an acceptable result. By contrast, the second alternative yields values that are close to the benchmark for all variables but one. For nine of the twelve key variables, the relative difference is less than 1%. It is interesting to note that automatic editing appears to have an adverse effect on the quality of the variable *Employed persons (FTE)*. This may be explained by the fact that the hard edits contain relatively little information about this variable: it is only involved in two inequality edits, whereas the other key variables are all involved in at least one equality edit.

The above results suggest that for this data set, some of the manual work could be replaced by automatic editing without affecting the quality of the main output. However, a more thorough analysis would be required before we can draw this conclusion. For one thing, we did not take the sampling design into account. Moreover, other quality indicators are important besides the unweighted means of key variables. The purpose of this analysis was merely to illustrate the effects of different editing strategies on real-world data.

6 Discussion and conclusions

In this paper we have discussed the relation between automated and manual (selective) data editing from three different viewpoints. The first viewpoint we take is that of the source of error. As is turns out, data editing staff spend considerable time editing data which are not observed survey data. Often, classifying variables from the business register (*e.g.* NACE codes) have to be altered as well. The source of error (overcoverage) is then not a measurement error of the survey but an error in the population register. The amendments proposed by editors in such cases are usually based on unstructured information such as web sites. Also, such amendments often have consequences for other statistical processes, for example when a centrally maintained variable (such as the NACE code) must be adapted.

The second viewpoint we take is from the current state of the art in automated data editing. The image emerging from the discussion and numerical examples

is that established automated methods tend to perform well for the majority of records, provided that hard edit rules have been defined and sufficient structured auxiliary information is available for the estimation of new values. Exceptions include mostly records of large businesses; these usually have a more complex structure than small establishments and data editing staff often uses external unstructured information to repair such records. Obviously, automated methods are better suited for (computationally, mathematically) complex calculations than data editing staff. On the other hand, data editing staff are better at judging the violation of soft edit rules, often again by using unstructured auxiliary information.

Thirdly, we discussed the relation between manual and automated data editing from the point of view of process design. We have decomposed the data editing process into several types of tasks (statistical functions), which are independent of how they are implemented: manually or automatically. This allowed us to separate the tasks which are currently easier to implement manually from those that may be implemented automatically. Here, we find that record selection, possibly supported by macro-editing tools, as well as judging and amending unit properties are often performed manually.

Table 4 summarises the above discussion. We may conclude that currently, automated methods serve very well to edit observed variables in business survey records of establishments that are not overly complex (large) and are restricted by hard edit rules. Automated methods are not yet suited for repairing records related to large, complex companies, records under soft restrictions or performing amendments based on unstructured data. Those tasks are still mostly performed manually. Of course, manual editing of observed variables of simple (small) units based on structured information is always possible; our point is that here the same quality can often be achieved more efficiently with automated methods.

The decomposition of data editing in different statistical functions given in Section 4 allows one to assess a data editing process on a task-by-task basis. This leads to a more refined complementation of automatic editing by (selective) manual editing than what emerges from the classical literature on selective editing. Evaluation of the results of automatic editing tasks also enables one to select the best automatic method for a specific task and thus minimise manual actions related to that task. Furthermore, the statistical functions in the decomposition each have their own set of minimal, well-defined inputs and outputs which are independent of the method used to implement the function. This modular approach to data editing offers clear potential for the development of reuseable components, yielding efficiency gains in process design.

To conclude, we see the following research opportunities. First of all, standard-

Table 4. Relative strengths and weaknesses of manual and automated data editing.

	Editing mode	
Aspect	Manual	Automated
Variable		
Observed variable	−	+
Unit property	+	−
Edit rules		
Hard edits	−	+
Soft edits	+	−
Use of aux. information		
Structured	−	+
Unstructured	+	−
Type of unit		
Simple	−	+
Complex	+	−

ised quality aspects of the statistical functions identified in Section 4 should be developed. Such aspects could be, for instance, the fraction of false negatives (or positives) in the selection of suspicious units or erroneous fields, the prediction accuracy of imputed values obtained by some imputation method or the reduction in bias of estimates due to different amendment functions. This, then, would allow of a standardised way to compare data editing processes and paves the way for further development of reusable components based on various methodologies. Secondly, data editing research should focus on areas where automated data editing is currently less suitable (Table 4). Interesting fields of research are the use of unstructured information to verify and/or amend data and the use of soft edits in automated data editing. Some recent progress in the latter field was made by one of the authors (Scholtus and Göksen, 2012; Scholtus, 2013). The use of for example web scraping or text mining techniques in data editing remains largely unexplored.

Acknowledgements

The authors are grateful to T. de Waal, M. Di Zio, U. Guarnera, I. Arbués, P. Revilla and D. Salgado for comments and suggestions and to L.-C. Zhang for fruitful discussions on the subject of this paper.

References

Al Hamad, A., D. Lewis, and P. L. N. Silva (2008). Assessing the performance of the thousand pounds automatic editing procedure at the ONS and the need for an alternative approach. Working Paper No. 21, UN/ECE Work Session on Statistical Data Editing, Vienna.

Albuéz, I., P. Revilla, and D. Salgado (2013). An optimization approach to selective editing. *Journal of Official Statistics current issue.*

Bethlehem, J. (2009). *Applied survey methods.* Wiley series in survey methodology. John Wiley & Sons, Inc.

Boskovitz, A. (2008). *Data editing and logic: the covering set method from the perspective of logic.* Ph. D. thesis, Australian National University.

Camstra, A. and R. Renssen (2011). Standard process steps based on standard methods as part of the business architecture. In *Proceedings of the 58th World Statistical Congress (Session STS044)*, pp. 1–10. International Statistical Iinstitute.

Damerau, F. (1964). A technique for computer detection and correction of spelling errors. *Communications of the ACM 7*, 171–176.

De Jonge, E. and M. Van der Loo (2011). Manipulation of linear edits and error localization with the editrules package. Technical Report 201120, Statistics Netherlands, The Hague.

De Jonge, E. and M. Van der Loo (2012). *editrules: R package for parsing and manipulating of edit rules and error localization.* R package version 2.5.

De Waal, T., J. Pannekoek, and S. Scholtus (2011). *Handbook of statistical data editing and imputation.* Wiley handbooks in survey methodology. John Wiley & Sons.

De Waal, T., J. Pannekoek, and S. Scholtus (2012). The editing of statistical data: methods and techniques for the efficient detection and correction of errors and missing values. *Wiley Interdisciplinary Reviews: Computational Statistics 4*, 204–210.

De Waal, T. and R. Quere (2003). A fast and simple algorithm for automatic editing of mixed data. *Journal of Official Statistics 19*, 383–402.

Di Zio, M. (2013). A contamination model for selective editing. *Journal of Official Statistics Current issue.*

Di Zio, M., U. Guarnera, and O. Luzi (2005). Editing systematic unity measure errors through mixture modelling. *Survey Methodology 31*, 53–63.

Fellegi, I. P. and D. Holt (1976). A systematic approach to automatic edit and imputation. *Journal of the American Statistical Association 71*, 17–35.

Granquist, L. and J. G. Kovar (1997). Editing of survey data: how much is enough? In L. Lyberg, P. Biemer, M. Collins, E. de Leeuw, C. Dippo, N. Schwartz, and D. Trewin (Eds.), *Survey measurement and process quality*, Wiley series in probability and statistics, pp. 416–435. Wiley.

Groves, R. M. (1989). *Survey errors and survey costs*. Wiley series in survey probability and mathematical statistics. John Wiley & Sons, Inc.

Hedlin, D. (2003). Score functions to reduce business survey editing at the U.K. Office for National Statistics. *Journal of Official Statistics 19*, 177–199.

Latouche, M. and J.-M. Berthelot (1992). Use of a score function to prioritize and limit recontacts in editing business surveys. *Journal of Official Statistics 8*, 389–400.

Lawrence, D. and C. McDavitt (1994). Significance editing in the Australian survey of average weekly earning. *Journal of Official Statistics 10*, 437–447.

Lawrence, D. and R. McKenzie (2000). The general application of significance editing. *Journal of Official Statistics 16*, 243–253.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady 10*, 707–710.

Little, R. J. A. and D. B. Rubin (2002). *Statistical Analysis with Missing Data* (second ed.). New York: John Wiley & Sons.

Pannekoek, J. and L.-C. Zhang (2011). Partial (donor) imputation with adjustments. Working Paper No. 40, UN/ECE Work Session on Statistical Data Editing, Ljubljana.

Pannekoek, J. and L.-C. Zhang (2012). On the general flow of editing. Working Paper No. 10, UN/ECE Work Session on Statistical Data Editing, Oslo.

Scholtus, S. (2009). Automatic correction of simple typing errors in numerical data with balance edits. Technical Report 09046, Statistics Netherlands, Den Haag.

Scholtus, S. (2011). Algorithms for correcting sign errors and rounding errors in business survey data. *Journal of Official Statistics 27*, 467–490.

Scholtus, S. (2013). Automatic editing with hard and soft edits. *Survey Methodology*. Accepted for publication.

Scholtus, S. and S. Göksen (2012). Automatic editing with hard and soft edits – some first experiences. Working Paper No. 39, UN/ECE Work Session on Statistical Data Editing, Oslo.

UNECE Secretariat (2009). Generic Statistical Business Process Model version 4.0. Joint UNECE/Eurostat/OECD Work Session on Statistical Metadata.

Van der Loo, M. (2012). *rspa: Adapt numerical records to fit (in)equality restrictions with the Successive Projection Algorithm.* R package version 0.1-1.

Van der Loo, M. and E. De Jonge (2011). Deductive imputation with the deducorrect package. Technical Report 201126, Statistics Netherlands.

Van der Loo, M., E. De Jonge, and S. Scholtus (2011). *deducorrect: Deductive correction, deductive imputation, and deterministic correction.* R package version 1.3-1.