WP.
ENGLISH ONLY

UNITED NATIONS
ECONOMIC COMMISSION FOR EUROPE

CONFERENCE OF EUROPEAN STATISTICIANS

Work Session on Statistical Data Editing
(Paris, France, 28-30 April 2014)

Topic (ii): New and emerging methods

IMPLEMENTATION AND EVALUATION OF AUTOMATIC EDITING

Prepared by Jeroen Pannekoek, Mark van der Loo and Bart van den Broek, Statistics Netherlands

I. INTRODUCTION

1. For business statistics, automatic editing is almost always an important part of the statistical production process. It often entails the application of a number of sub-tasks or editing functions, each with their own purpose and configuration requirements (e.g. edit-checking, localisation of random and systematic errors, imputation of missing or discarded values, adjustment of values for consistency). The cost-effectiveness and transparency of designing, implementing and maintaining automatic editing systems can be greatly improved by the use of standardised re-usable methodology and tools. Current work on automatic editing at Statistics Netherlands is targeted at identifying generalisable standard data editing functions, supported by documented standard methods and implemented in R-based tools. This leads to a modular approach where the overall data editing process is decomposed in a number of standard re-usable process steps that connect in a plug-and-play manner. In this paper we will discuss the implementation of such modular systems and also the application of indicators that can measure the effect of each process step. Graphical displays may be used to allow for a concise review of the progress of the process as it proceeds according the the different process steps. This monitoring can provide continuous feedback on the quality of the data and the methods and parameters of the data editing system, which facilitates the integration of process optimisation as a part of the standard production process.

2. This paper is organised as follows. In section II we discuss the decomposition of an overall automatic editing process in process steps that can individually be monitored and evaluated. Section III presents views on changes to the data across the process steps and in section IV we discuss views on edit rule violations across process steps. In section V a few conclusions are summarised.

II. DATA EDITING PROCESS STEPS

A. Statistical functions and process steps

3. To design, evaluate and optimise the data editing processes in an efficient and generalisable way, the process is decomposed in process steps which can be applied, as much as possible, with generalised methods and software tools. In addition the process steps should also be defined such

that a step-by-step monitoring of the process supports the optimisation of parameter settings and comparison of alternative methods.

4. Pannekoek et al. [2013] describe 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. This decomposition is shown in Figure 1. The data editing tasks are decomposed, 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 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 or frame variables. This may be interpreted as a decomposition based on a record-wise or field-wise action.

5. 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 task (a process step) a method for that function must be specified and configured. The same statistical function can, and often will, be implemented by several methods even within the same application. For instance, a number of different methods for detecting erroneous fields will often be applied one after another so as to catch as many errors as possible. This may be seen as the repeated application of the function *field selection*.

6. The statistical functions defined here each have their own minimal input-output specification which is independent of the chosen statistical method or implementation thereof. For record-wise verifications, the output is an $N \times K$ matrix (see section IV), with N the number of units and K the number of functions (edit rules or score functions) to be evaluated. In case of scoring the $N \times K$ matrix holds the *local scores* from which, by an operation on the rows of this matrix, the N-vector with *record scores* or *global scores* is obtained. Field selection functions have indicator values for each cell of the $N \times J$ data matrix as their output, with J the number of variables. Record selection functions have an N-vector of indicators as their output (for instance indicating whether the record scores are above a cut-off value, as in selective editing). Amendment functions actually change data or unit properties, they have data or unit properties as input and revised data or unit properties as there output. The effects of amendment functions can be measured by (functions of) the difference between the data before and after amendment (see section III).

7. An actual implementation of a data editing process can now be seen as a collection of implementations of statistical functions (process steps). 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, see Pannekoek and Zhang [2012] for a discussion of these choices. As an example of such a process, we have listed in table 1 the process steps that were used in an automatic editing system for data on childcare institutions and that will be used as one of the examples in this paper. The process steps implement selection and amendment functions with different methods. Some methods are based on generalised methods, models and/or algorithms, while others use only simple direct if-then type of rules. The first five steps are all correction steps for errors with a detectable cause, they combine a field selection function (detection of a specific kind of error) with an amendment function (correction of that type of error). They are performed by direct rules

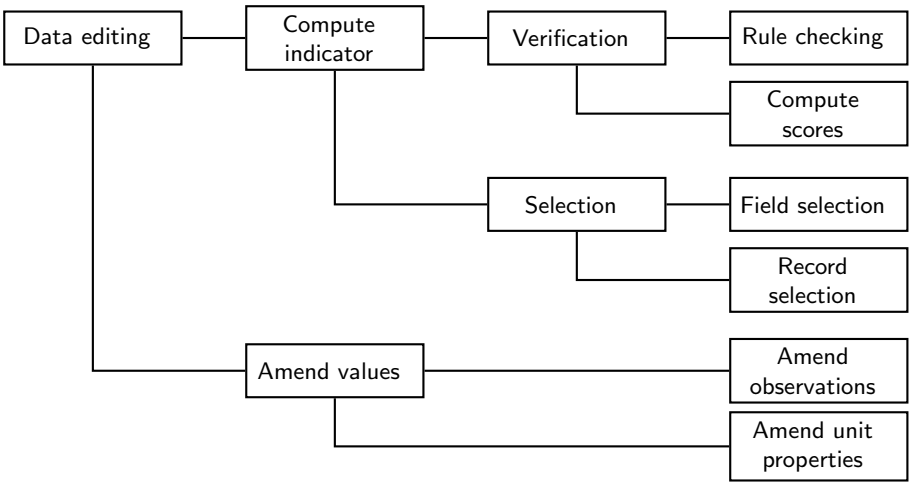

FIGURE 1. A taxonomy of data editing functions.

TABLE 1. Relation between error localisation and amendment process steps, software components and rule sets for the automatic editing of the data of Childcare Institutions.

Process step	**Software routine**	**Rule set**
1. Correction with direct rules	applyRules	Correction rules
2. Thousand error correction with rules	applyRules	Thousand error rules
3. Correction of typos	correctTypos	Edit rules
4. Correction of rounding errors	correctRoundings	Edit rules
5. Error localisation with direct rules	applyRules	Error loc. rules
6. Error localisation under the FH paradigm	localizeErrors	Edit rules
7. Deductive imputation of implied values	impliedValues	Edit rules
8. NN-imputation	custom	
9. Adjustment of imputed values	adjustValues	Edit rules

(different kinds of corrections, including incorrect minus signs, in step 1; thousand errors in step 2) and by generalised methods and algorithms (typo's and rounding in step 4 and 5). Step 5 and 6 are field selection steps (localise errors) by, respectively, direct rules and a general algorithm based on the paradigm of Fellegi and Holt [1976]. Steps 7, 8 and 9 are amendment functions, they have the effect that missing or erroneous and therefore discarded values are imputed with new values (step 7 and 8) and that these imputed values are adjusted, as little as possible, to ensure consistency with all edit rules. These last three steps all use generalised methods or algorithms. A detailed account of many methods and algorithms that can be applied in each of these steps can be found in De Waal et al. [2011].

8. The most important part of the configuration of the steps in table 1 is the set of rules that is used by the methods (mentioned in the last column). For correction and localisation steps with direct rules, these rule sets consist obviously of the if-then type of rules themselves. For the other steps the edit rules determine the result. So, there are four different sets of rules used by nine process steps and, as can be seen from the second column, executed by seven different software routines, six of which are generalised standard re-usable software components. The imputation routine was custom build, but easy to implement. After each process step the changes to the data set are saved so that the effects on the data can be monitored. For a review (and tutorial) of the standard software routines we refer to de Jonge and van der Loo [2013] and the references cited there.

9. The verification with hard edit rules is implicitly applied in a number of process steps, in fact all steps in table 1 for which the rule set is "Edit rules". They can, however, also be applied after each process step to monitor the effects of that step in terms of these pre-specified validation rules and the same holds true for soft edit rules and score functions, see section IV.

10. An overall data editing process can be split up in different process steps in many ways. For instance, from a technical point of view the first two process steps in table 1 could be combined by merging the rule sets and using the routine applyRules just once with this combined rule set, without altering the results. However, for the purpose of evaluating the effects of applying the rules it is much more informative to split the rules in those for correcting thousand errors and other direct correction rules.

B. Example data sets

11. To illustrate the effects of applying a sequence of automatic editing steps we will use, in this section and the next, two examples. The data for the first example are a subset of 840 records and 76 variables taken from a census among institutions for child day care. The variables concern the production, costs, revenues and personnel and are similar to what is typical for structural business surveys. For these variables 78 hard edit rules have been specified. The second example is a data set of 323 records from the Dutch SBS. The data are on businesses with ten employees or more from the sector wholesale in agricultural products and livestock. This survey contains 93 variables that should conform to 120 linear edits of which 19 are equalities. The valid values for the data sets are defined by edit rules which specify the admissible values in terms of linear equalities and inequalities, e.g. *profit+ total costs - turnover = 0* and *total costs = employee costs + costs of purchases + other costs* or *number of employees* $\leq$ *employees in FTE.*

12. The process steps for the childcare data have been shown in table 1. The editing steps for the wholesale data are similar but slightly different. They start with three correction steps using direct rules: 1. correction for incorrect minus signs, 2. correction for thousand errors and 3. other corrections with direct rules. Then three steps that are based on algorithms: 4. typos, 5. rounding and 6. FH-error localisation. Then, for e few selected records a manual editing step was performed (7). The last three steps involve imputation and adjustment: 8. deductive imputation, 9. regression imputation and 10. adjustment of imputed values.

III. VIEWS ON DATA VALUES DURING THE CORRECTIION PROCESS

A. The status of values of variables

TABLE 2. Data cells classified by their status in the editing process.

Total number of cells				
available			missing	
still available		made available (imputed)	still missing	made missing (cancelled)
available, unaltered	available, amended			

13. During the editing process, the values in in the cells of the data matrix, may be changed. At each point in this process we can assign to each cell a status that reflects how the value of that cell has or has not changed with respect to a previous state of the process. To this end we define a status variable with five categories, as shown in table 2. The categories of the cell status are obtained by first dividing the cells in cells with available values and cells with missing values, then dividing the cells in the available category in those that where available in the previous state and remain so (still available) and those that have become available by imputation. The still available cells can be divided further in cells with values that remain the same (unaltered) and cells with amended values. The cells with missing values can be subdivided in those that were also missing in the previous state (still missing) and those for which the value has been made missing (canceled), because the value has been detected as erroneous and set to be missing to be imputed later on. A similar description of the cell status in relation to editing and imputation was proposed by Della Rocca et al. [2005].

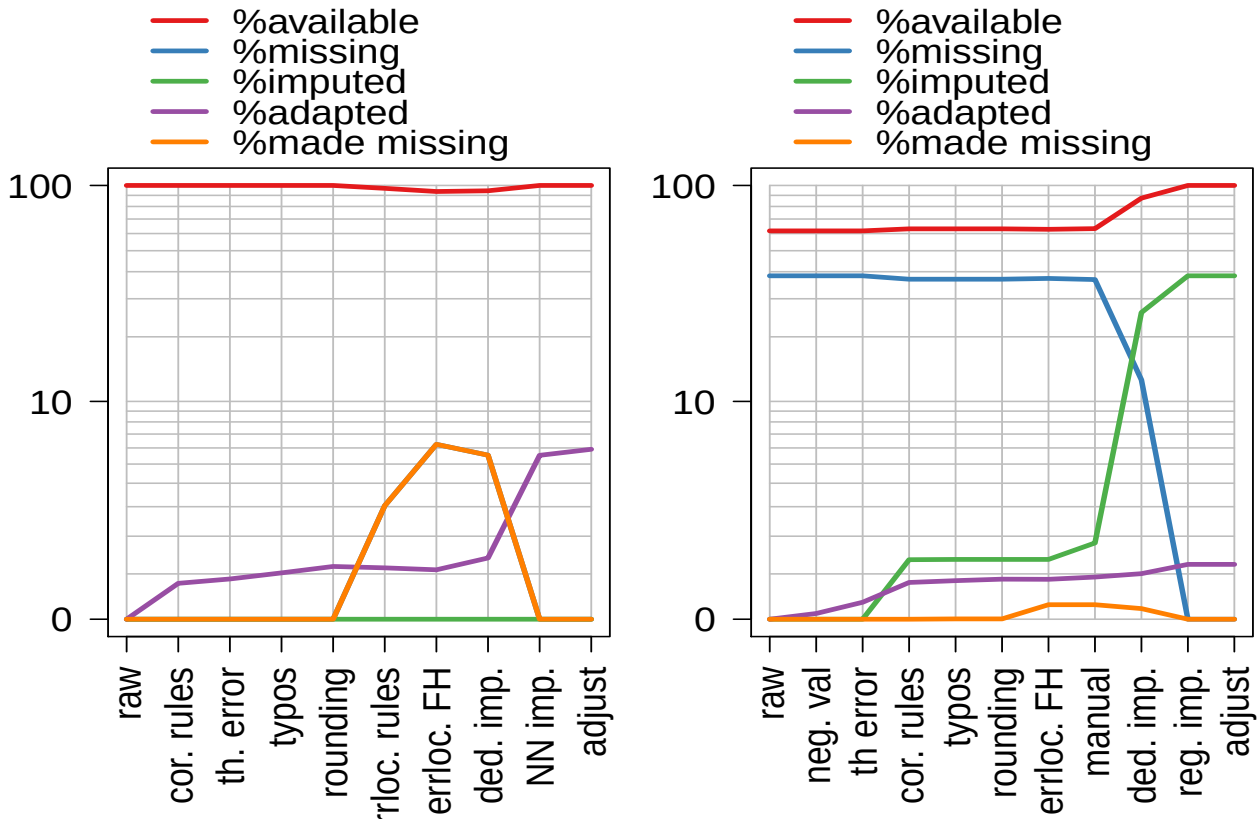

FIGURE 2. Cell status changes for Childcare institutions (left) and Wholesale (right). In percentages on a pseudo log-scale.

14. A graphical representation of the changes in cell status by process step, relative to the raw data, for the childcare institutions and wholesale data sets is given in figure 2. For the childcare data it was customary to fill in blanks by zeros and therefore the number of missing values starts at zero while for the wholesale data there is a considerable amount of missing data to start with. Although this difference may be somewhat artificial because of the filling in with zeros, it highlights the effects of the amount of missing values on the whole editing process. For the childcare data much more values are made missing by the error localisation steps while for the wholesale data error localisation does not have such a large effect because for edit rules that already contain missing values it is unnecessary to set more values to missing to resolve an edit failure. The percentage of imputed values refers to

values that are missing in the raw data and filled in with some value during the correction process. This is zero for the childcare data but it is the main change to the data cells statuses for the wholesale data. Imputation does play an important role for the childcare data as well but here it appears as the percentage adapted because the value of a non-missing cell in the raw data is changed, by first setting it to missing by error localisation and then imputing a new value.

B. Means and variances

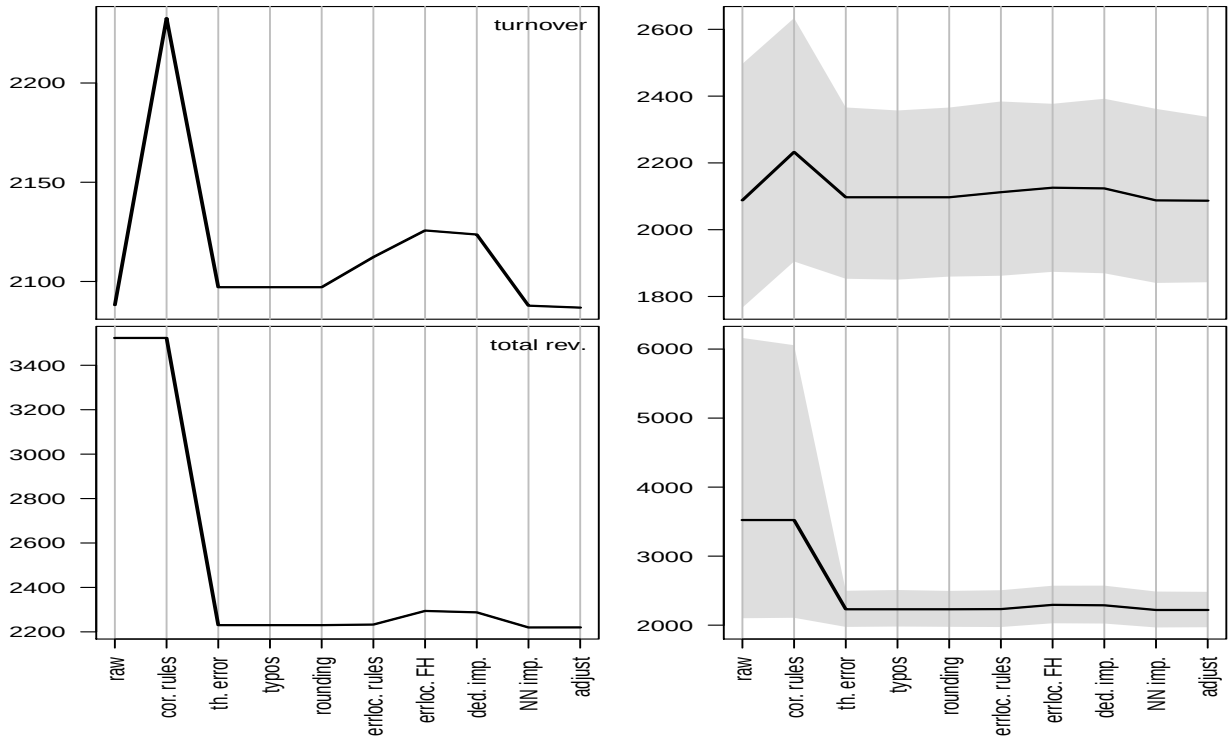

FIGURE 3. Changes in means and confidence intervals for Turnover and Total revenues, Childcare institutions.

The effects of data editing actions on estimates can be monitored by plotting the means of important variables against process steps. This is illustrated in figure 3 for the childcare data. This figure shows the means for the variables *turnover* and *total revenues* (left side) at each process step and associated estimated 95% confidence intervals (right side) based on the data at that process step. Correction with direct rules has a relatively large effect on *turnover* but no effect on *total revenues*. The correction for thousand errors decreases the mean for both these variables but the effect for *total revenues* is much larger. For *total revenues* we also see that the estimated confidence interval becomes much smaller when the thousand errors are removed. Error localisation increases the means for both variables, which shows that smaller values are more often localised as errors than larger ones: the errors occur not at random. After imputation the means decrease again, showing that the imputed values are smaller than the observed ones they replace.

IV. VIEWS ON EDIT RULES DURING THE CORRECTION PROCESS

A. The verification status of edit rules

15. The linear hard edit rules we consider here are of the general form $\mathbf{a}^T\mathbf{x} = b$, for equalities or $\mathbf{a}^T\mathbf{x} \leq b$, for inequalities, with $\mathbf{x}$ the vector with variables in a record, $\mathbf{a}$ a vector of the same length with constants and b a scalar constant. For the examples in section II.B, b=0 and $\mathbf{a}$ consists of zeros, ones and minus ones. Alternatively, such rules can be expressed as $\mathbf{a}^T\mathbf{x} \in V$, where V is the set containing the admissible (Valid) values for $\mathbf{a}^T\mathbf{x} - \mathbf{b}$, that is V is equal to the single element 0 for equalities and to the interval $(-\infty, 0)$ for inequalities. Edit checking, then, involves evaluating $\mathbf{a}^T\mathbf{x}-\mathbf{b}$, or more generally a function $s_e(\mathbf{x})$ that returns the value to be checked, and comparing the result with the valid-values set V. A complication arises if $\mathbf{x}$ contains missing values so that $s_e(\mathbf{x})$ cannot be evaluated. If the missing values occur in variables that are not contained in the edit, corresponding to the zero values in $\mathbf{a}$ for linear edits, they are irrelevant for evaluating the edit and can be ignored, which can be accomplished (for linear edits) by introducing the convention 0.NA=0. For missing values in variables that *are* contained in the edit, the problem remains and the edit cannot be evaluated. The checking of a record $\mathbf{x}$ against an edit e now results in a three-valued function:

$$e(\mathbf{x}) = \begin{cases} \texttt{TRUE} & \text{if } s_e(\mathbf{x}) \in V \\ \texttt{FALSE} & \text{if } s_e(\mathbf{x}) \notin V \\ \texttt{NA} & \text{if } s_e(\mathbf{x}) \text{ cannot be evaluated.} \end{cases} \tag{1}$$

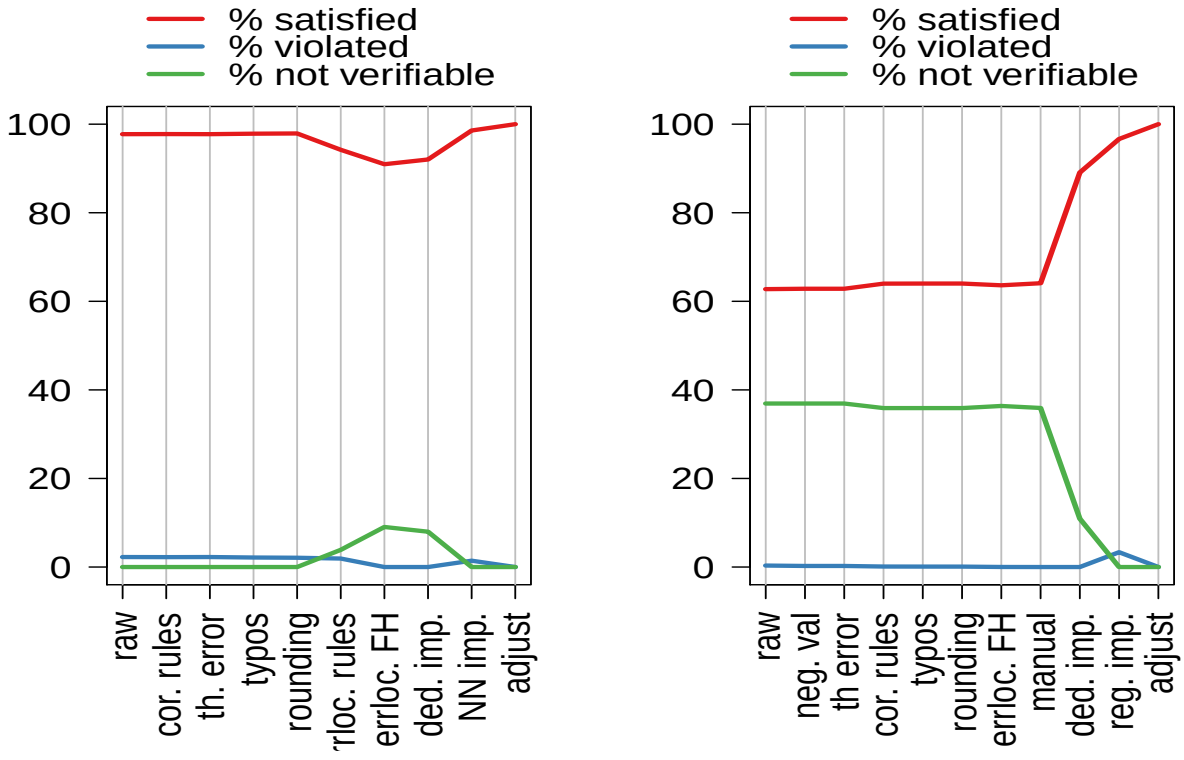

FIGURE 4. Edit verification status, Childcare institutions (left) and Wholesale (right).

16. By checking each record against each edit rule, we obtain the $N \times K$ failed edit indicator matrix $\mathbf{F}$ with elements $e_k(\mathbf{x}_i)$, for $i = 1, \cdots, N$ and $k = 1, \cdots, K$. This matrix with quality measures

can be summarised and analysed in several ways: numbers of failures by edit, by variable or by process step or combinations of these. Figure 4 shows the percentages of failures by process step for the childcare and wholesale data. This figure highlights a number of differences between these two statistical processes. For the childcare data, because of the filling in of blanks by zeros, the number of not verifiable edits starts at zero while for the wholesale data the effect of missing data is that roughly 40% of the edits cannot be evaluated. Error localisation for the childcare data has the effect of reducing, as intended, the number of violated edits but it also reduces the number of satisfied edits by changing their status to "not verifiable". This can be explained by the fact that many variables are contained in several edits, and if a variable is set to missing all edits containing this variable will become unverifiable, including those that where satisfied. For the wholesale data the main problem is the many missing values, error localisation hardly adds to the number of missings, and the imputation methods have the largest effects on the numbers of satisfied an verifiable edits. While deductive imputation cannot increase the number of violated edits, regression imputation does.

B. Edit tolerances and score functions

17. If an edit rule is violated, we have more information than its verification status, we can also consider the amount by which the rule is violated. We call this the edit *tolerance*. This concept is similar to Hedlin's concept of edit-related score functions (Hedlin [2003]), which is based on the idea that the *amount* of failure of an edit can be used in a score for selective eding, but the tolrance defined below is more general.

18. As a measure of the tolerance of a record $\mathbf{x}$ for an edit e, we consider the shortest distance (by some measure $D(.,.)$) between the failing record $\mathbf{x}$ and the set of records satisfying e, that is, we define the tolerance as

$$t(\mathbf{x}, e) = \begin{cases} 0 & \text{if } s_e(\mathbf{x}) \in V \\ \min_{\mathbf{y}}(D(\mathbf{x}, \mathbf{y}) : \mathbf{y} \in V) & \text{if } s_e(\mathbf{x}) \notin V \\ \texttt{NA} & \text{if } s_e(\mathbf{x}) \text{ cannot be evaluated.} \end{cases} \tag{2}$$

The value of $\mathbf{y}$ obtained by the minimisation in (2) can be thought of as a minimally adjusted version of $\mathbf{x}$ such that $\mathbf{y}$ satisfies e *and* the difference between $\mathbf{y}$ and $\mathbf{x}$ as measured by $D(\mathbf{x}, \mathbf{y})$ is minimal. The tolerance, then, is the value of $D(\mathbf{x}, \mathbf{y})$ corresponding to this minimum, it is a measure of the amount of change in $\mathbf{x}$ necessary to satisfy e.

19. For D the Euclidean distance and linear edits, the tolerance value for $s_e(\mathbf{x}) \notin V$ can be obtained by solving the minimisation problem $\min_{\mathbf{y}} ||\mathbf{x} - \mathbf{y}||$ subject to $s_e(\mathbf{y}) \in V$, for which the solution is given by (van den Broek et al. [2014])

$$t(\mathbf{x}, e) = (\mathbf{a}^T\mathbf{a})^{-1}|\mathbf{a}^T\mathbf{x} - b| = (\mathbf{a}^T\mathbf{a})^{-1}|s_e(\mathbf{x})|, \tag{3}$$

which is intuitively plausible since it is a constant times the difference between $\mathbf{a}^T\mathbf{x}$ and the required (for equalities) or maximal admissible (for inequalities) value b.

20. The hard edit tolerance can be seen as an edit-related score function. It can readily be calculated from the hard edit rules that are typically available in automatic editing of business records. However, the more commonly applied estimate-related score functions (see e.g. Hidiroglou and Berthelot [1986], Lawrence and McKenzie [2000]) also fit into the framework of evaluating process steps by the value of a function that indicates the implausibility of values in a data record. Let $s_d(\mathbf{x})$ be a local score function, for instance a function that compares the current value of a variable or ratio's of variables with historical values or current medians. Then we can calculate for each record $\mathbf{x}$ the value

for each such function $s_d(\mathbf{x})$ if the values of the variables involved in s_d are not missing, thus we have

$$t(\mathbf{x}, d) = \begin{cases} s_d(\mathbf{x}) & \text{if } s_d(\mathbf{x}) \text{ can be evaluated} \\ \texttt{NA} & \text{if } s_d(\mathbf{x}) \text{ cannot be evaluated,} \end{cases} \tag{4}$$

where large values of $t(\mathbf{x}, d)$ point at values that are implausible according the local score s_d at the process step that is evaluated.

21. In figure 5 boxplots are shown of the distribution of edit tolerances for the wholesale data at each process step. This figure shows an increase of the median of the tolerances for the correction for negative values, which is reason for a more detailed examination of the changes made in this step. Correction of thousand errors hardly effects the tolerances since uniform thousand errors tend not to break edit rules. A substantive decrease of the median as well as the number of non-zero tolerances is the result of the correction by direct rules. Error localisation by the Fellegi-Holt paradigm reduces the number of non-zero tolerances almost to zero but increases the number of not evaluated tolerances. In principle this step should leave no non-zero tolerances at all, but for a view records the older implementation of the FH-algorithm didn't converge and the error localisation problem was not solved (in our current R-software this problem doesn't occur). Deductive imputation cannot result in edit violations but regression imputation does result in many non-zero tolerances, but the number of tolerances that can be evaluated is also substantially larger than in previous steps. After adjustment of the imputed values the data are consistent with all edit rules and all tolerances are zero.

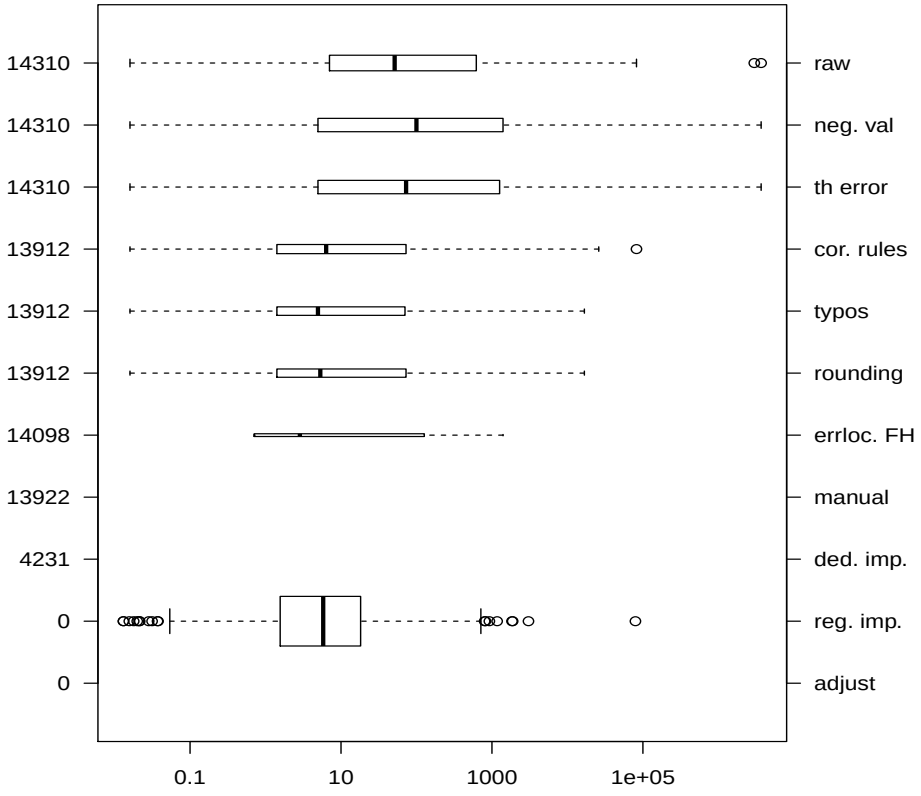

FIGURE 5. Boxplots of edit tolerances for the wholesale data. The height of the boxes is proportional to the square root of the number of non-zero tolerances. On the left-hand side are the numbers of not evaluated tolerances.

V. CONCLUSIONS

22. Although automatic editing can be implemented as a black box where only the input and final output are visible for the users, this ignores the detailed information that can become available by evaluating the effects of each step of the automatic data editing process. This information can reveal which kinds of errors were detected and how much correction was needed to make the data conform to the different (sets of) rules that drive the automatic editing system. This detailed step-by-step monitoring can reveal unexpected effects of the data editing system and locate by which methods(s) or rule(s) these effects were caused.

23. We considered measures to evaluate the process over the process steps, among which are score functions that are traditionally used to single out records with influential but suspect values. In the context of this paper these scores are used to measure the effects of each automatic editing step on the amount and size of suspect and deviating values. Apart from the usual estimate related score-functiosn we also considered a generalisation of edit-related score functions that can be calculated using the hard edit rules that are already available in the automatic editing system. This information can be used to either review the methods and rules that drive the automatic editing process or by branching, possibly at each step, to an efficient selective editing step in which selected data values are manually reviewed.

References

E. de Jonge and M. van der Loo. An introduction to data cleaning with R. Technical Report 201313, Statistics Netherlands, 2013. URL http://www.cbs.nl/nl-NL/menu/methoden/onderzoek-methoden/discussionpapers/archief/2013/default.htm.

T. De Waal, J. Pannekoek, and S. Scholtus. *Handbook of statistical data editing and imputation*. Wiley handbooks in survey methodology. John Wiley & Sons, 2011. ISBN 978-470-54280-4.

G. Della Rocca, O. Luzi, M. Signore, and G. Simeoni. Quality indicators for evaluating and documenting editing and imputation. Working paper No. 3, UN/ECE Work Session on Statistical Data Editing, Ottawa, 2005.

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

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

M. A. Hidiroglou and J. M. Berthelot. Statistical editing and imputation for periodic business surveys. *Survey Methodology*, 12:73–78, 1986.

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

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

J. Pannekoek, S. Scholtus, and M. van der Loo. Automated and manual data editing: a view on process design and methodology. *Journal of Official Statistics*, 29:511–537, 2013.

B. van den Broek, M. van der Loo, and J. Pannekoek. Kwaliteitsmaten voor het datacorrectieproces. Technical Report 201408, Statistics Netherlands, 2014. URL http://www.cbs.nl/nl-NL/menu/methoden/onderzoek-methoden/discussionpapers/archief/2014/default.htm.