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 (v): International collaboration and processing tools

TOWARDS GENERIC ANALYSES OF DATA VALIDATION FUNCTIONS

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

I. INTRODUCTION

1. The ability to quantitatively or qualitatively assess the effect of a data processing activity is fundamental to statistical production. Given the abundance of information that can in principle be mined from complex, multi-step data processing activities, having access to indicators that summarize that information in a meaningful way useful at all levels of a data processing organisation. Indeed, Brancato et al. [2009], show how various indicators may be used to report to different actors including survey managers, quality managers and external users. Furthermore, Pannekoek et al. [2014] show a number of indicators that can be meaningfully applied to follow impact of process steps on business survey data.

2. The current trend towards the use of data sources (administrative, web-based) for which the data generation process is beyond the control of an NSI, has consequences for the ensuing statistical production process. In particular, one may expect that production processes need to be more flexibly (re)assembled to cope with variations in input data. Ideally, one imagines a process where standard tools are interactively assembled into a production system while effects on data can be monitored in real time. Currently, there are several (international) standardisation initiatives on their way that are relevant to tool-building. These activities include the development of generic information models (GSIM), data exchange formats (DDI, SDMX) and more. Besides a high degree of standardisation in tool design, it is beneficial to be able to flexibly define and apply validation methods to data at all stages of production. Generic analyses of validation output that can be applied to any validation method is especially beneficial since it allows for comparison of methods and tools.

3. In this paper we discuss the relation of data validation activities with several (upcoming) international standards. Next, we give a precise definition of data validation in terms of a three-valued function, which can be interpreted as a composition of a score function and a decision function. Based on this definition we derive two generic parameters that can in principle be determined for any data validation method (under a few assumptions). Finally, we show how these parameters for a number of common types of validation rules can be derived.

4. The work described in this paper is still ongoing and builds upon several recent works in which the authors have been involved. In Pannekoek et al. [2013] we have described a taxonomy of types data editing activities (called data editing *functions*, see Figure 1) that abstracts away from any particular

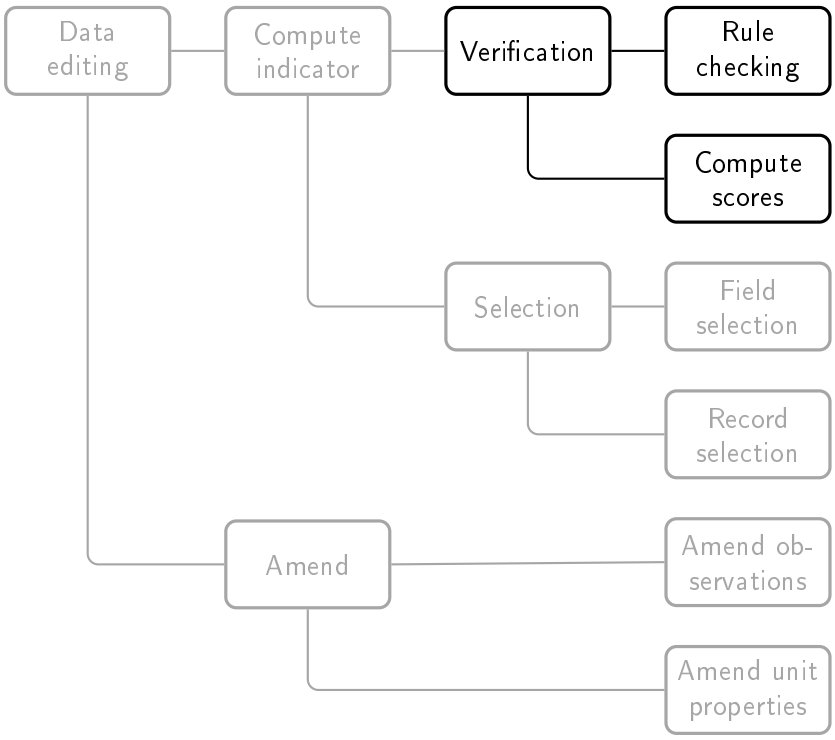

FIGURE 1. Taxonomy of data editing activities of Pannekoek et al. [2013]. In this paper we zoom in on data verification (also called data validation) functions.

chosen methodology. Data validation (or verification) is one of those fundamental activities. Each fundamental activity has similar type of output, and in this work we work further on specifying the output of validation activities to make such activities generically analyzable. Secondly, we have recently developed a number of data- and data processing quality indicators (reported upon in Pannekoek et al. [2014]) which enable us to follow the state of a data set as it gets processed for data editing to a certain extent. Finally, the immediate cause for this work is that we are looking for ways to implement more generic validation rules as an extension of our previously released data editing software based on the R environment for statistical computing. See de Jonge and van der Loo [2013] for a recent overview.

5. The rest of this paper is organized as follows. In the next section we shortly discuss our taxonomy and the activity of data validation in relation to current and upcoming international standards. In Section III we present a generic model for validation functions and work out some practical examples for cases of univariate, in-record and cross-record validation rules. Conclusions are given in Section IV.

II. Relation with international standards

6. There are currently several international (upcoming) standards that aim to facilitate a modular and flexible approach to designing and building statistical production systems. Here, we very briefly review their relation with the activity of data cleaning.

7. The generic statistical business process model (GSBPM) is a high-level classification of activities performed at official statistics institutes. The current classification (version 5.0, UNECE [2013a])

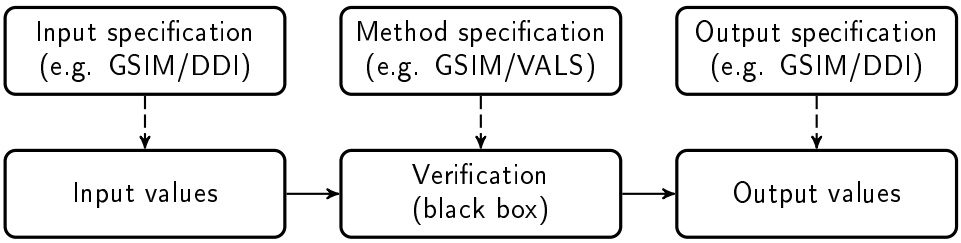

FIGURE 2. Specification of a validation step. In this paper we are not concerned with the specification of the method. The abbreviations in brackets are (examples of) standards that relate to that part of the specification.

consists of two levels where the first level divides activities into eight classes, starting with 'specify needs' and ending with 'evaluate'. Data processing is the fifth class. This class is again subdivided into eight types of activities of which '5.3: Review and validate' and '5.4: edit & imputation' are relevant for the current paper. These two classes can be mapped respectively to 'Verification' (or validation) and the combined classes 'Selection' and 'Amend values' in the taxonomy of Figure 1.

8. The generic statistical information model (GSIM) defines a set of information objects that play a role in official statistics production. GSIM version 1.1 [UNECE, 2013b] defines five groups of information objects, relating to for example processes (the business group) or data (the concepts group). Each group consists of a list of definitions of terms such as 'data', 'data point', 'unit', and 'population'. It furthermore defines the relation between those terms. For example, a population *contains* a set of units. There also exist connections between terms defined in different groups so the groups themselves are connected as well. Of relevance to us is the specification of input and output concepts in terms of GSIM objects, as well as method specification in terms of validation rules (see Figure 2). In terms of GSIM objects, the input and output of a validation process are simply 'Data sets', where data may either relate directly to observed values (the input) or be derived thereof (the output). The prescription of a validation method consists of a 'Process method' including (mathematical) validation 'Rules'

9. The common statistical production architecture (CSPA) documents a reference business architecture that allows activities at GSBPM level to be offered as a service within or across statistical organisations. CSPA version 1.0 [UNECE, 2013c] defines roles and tasks in an organization and offers templates for defining service interfaces. The work described in this paper may be of some relevance when implementing service interfaces as templated by CSPA.

10. While the GSBPM and GSIM are conceptual standards that abstract away from any implementation, technical standards can be used to describe a validation step and its in- and output in practice. The Data Documentation Initiative [2014] (DDI) for example, prescribes an XML-based standard for storing metadata throughout the life-cycle of a dataset. It is originally aimed at practitioners in social sciences but has gained some popularity amongst NSI's as well. The statistical data and metadata exchange format (SDMX, [SDMX working group, 2014]) is being developed by international governmental organisations including Eurostat, the Organisation for Economic Development and the European Central Bank. At the moment, European NSI's use it to exchange census data, for example. Both DDI and SDMX should in principle be capable of describing in- and output data. It is beyond the scope of this paper to compare them but recently a comparison was made as part of the ESSnet on SDMX2 [2013]. It is noteworthy that besides the XML-based standards prescribed by DDI and SDMX, standards like JSON and Protocol Buffers are currently widely in use.

11. The validation syntax (VALS) is a syntax definition that is still under development. It allows users to define validation rules for in-record, cross-record, and cross-dataset validation rules. Version 0.1309 Simon [2013] allows users to define validation rules in the form of boolean expressions which may include simple arithmetic or mathematical functions. The result of a VALS validation consists of a boolean validation result, a discrepancy value measuring the discrepancy between the actual data and data obeying the validation rule, and a severity measure indicating how 'serious' a violation of the rule at hand must be taken. Besides the validation rule, both the severity and discrepancy measure may be user-defined.

III. A model for validation functions

12. With validation, or (verification), we mean a confrontation of data with a previously defined quality requirement. Depending on the type and dimensionality of data (uni- or multivariate, single- or multirecord, structured or unstructured), the types and dimensions of input may vary wildly. Therefore, in the following we shall assume that the data that is being validated has already undergone a certain level of technical processing. Specifically, we assume that the data items are either empty or stored conforming the intended data model: numbers are stored as numbers, text as text and for categorical variables, each value is either empty or an existing (but possibly erroneous) category.

13. For such data, we define a *validation function* as as follows.

$$v(x) = \begin{cases} 1 \text{ if } s(x) \in V \\ 0 \text{ if } s(x) \notin V \\ \mathsf{NA} \text{ if } s(x) \text{ cannot be determined.} \end{cases} \tag{1}$$

Here, x is a data item, and we say x satisfies v when $v(x) = 1$ and x violates v when $v(x) = 0$. For the moment, we leave undetermined whether x represents a single value, a record or a data set; specific cases will be treated below. The function s is called a *score* function and V a region of the image of s whose values are considered valid. The value NA stands for 'not available' and occurs for example when (part of) x is missing and the value of $s(x)$ cannot be determined.

14. A validation function is thus fully specified once the score function and its valid region are defined. Observe furthermore that the validation function can be written as the function composition

$$v = i_V \circ s, \tag{2}$$

where i_V is the set indicator returning 1 when $s(x) \in V$, 0 when $s(x) \notin V$ and NA when $s(x)$ cannot be computed. Based on this composition we find two interesting cases. First, observe that the function $s(x)$ corresponds to the 'Compute scores' edit function of Figure 1. So leaving out the set indicator i_V from Equation (2) yields a quality indicator whose value may be interesting in its own right. See for example Pannekoek et al. [2014]. Second, if we set $s = \mathrm{Id}$, the identity function, the validation function reduces to a logical rule, restricting x to a subset of all possible values for x.

15. The definition of Eq. (1) suggests two measures of mismatch between actual data and data obeying all validation requirements. First, if we have a distance function d that quantifies the difference between two data points x and x' then the value

$$I(x, v) = \inf\{d(x, x') : v(x') = 0\}, \tag{3}$$

is the shortest distance from x to an x' that satisfies v (recall that the 'inf' operator takes the greatest lower bound of a set). The value of I may be interpreted as as measure of impact or influence of a particular violation on statistics based on x. We shall therefore refer to I as the *impact function*.

In general, finding the actual value is not trivial since it directly depends on the (possibly complex) definition of the chosen distance function d and the validation function v. In particular, one needs to be able to invert the score function at the boundaries of the valid region. However, below we will show that for a number of familiar validation rules it can be computed with relative ease.

16. Secondly, if we have a distance function on the image of s, the function

$$R(x, v) = \inf\{d(s(x), s') : s' \in V\} \tag{4}$$

measures how much the value of the score function needs to change before a x satisfies v. In many cases, this distance will be easier to compute, for example when both $s(x)$ and V consist of a single number, d may be just the absolute difference between the two.

17. It is clear that the functions I and R are often closely related: score functions are often chosen in a way to reflect the impact a violation will have on a statistic. An increase in I often yields an increase in R by design. In fact if we choose $s = \mathrm{Id}$, we see that I and R are identical. Therefore, if I is interpreted as the direct impact of a violation, then R may be interpreted as the severity of a violation: a higher R-score means a more severe violation. By moderating the score function we may quantitatively express the value we attach to a certain violation. The product of R and I can then be seen as a function that prioritizes treatment: an x yielding a higher priority value gets higher (manual) treatment priority.

18. We thus see that a validation activity may generically produce three types of output: the boolean value of the validation function $v(x)$, the impact function I and the severity function R, where the latter two each require the definition of a distance function. The impact function requires that distance between data points can be measured while the severity function requires that difference between scores can be measured. Each output type is defined irrespective of particular validation requirements. It is noteworthy that the VALS syntax mentioned above also defines three types of output (a boolean, and optionally a user-defined discrepancy and severity). The main difference with the current discussion is that here, the three values may be derived directly from a boolean validation rule. In the next subsections we work out a number of familiar examples.

A. Univariate, in-record validation

19. These validations can be executed by comparing a (function of) single values with an allowed set of values, irrespective of other values in the record or data set. For example, for a numeric value y we may have a range edit stating $y \geq b$, with b some chosen constant. To specify this rule in terms of Equation (1) we need to determine the score function s and the region of valid outcomes V. Here, we have

$$s(y) = y - b \text{ and } V = [0, \infty). \tag{5}$$

The reader may check that under this definition we have for example $v(1) = 0$, $v(-1) = 1$ and $v(\mathsf{NA}) = \mathsf{NA}$. Furthermore, if we choose the distance function $d(y, y') = |y - y'|$, we have

$$I(y, v) = R(y, v) = \begin{cases} 0 \text{ if } y - b \geq 0 \\ |y - b| \text{ if } y < b \\ \mathsf{NA} \text{ if } y = \mathsf{NA}. \end{cases} \tag{6}$$

Here, both I and R directly measure the amount of change in data necessary to satisfy the range edit.

20. As a more complicated example, consider a univariate outlier detection method based on a method introduced by Hiridoglou and Berthelot [1986]. Here, one computes the value of $f_{\mathsf{hb}}(y)$, given

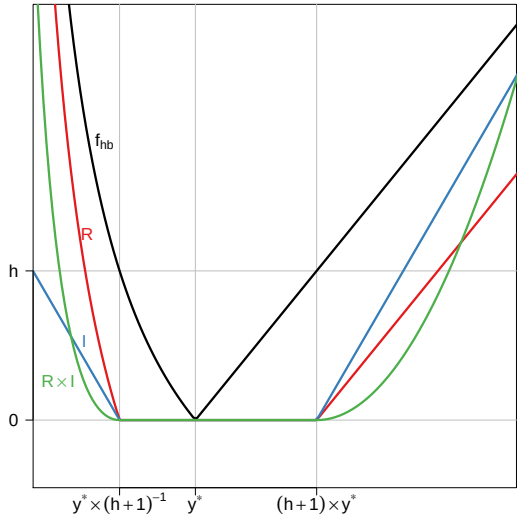

FIGURE 3. Schematic plot of f_{hb} [Eq. (7)], the corresponding R and I functions [Eqs. (8) and (9)], and prioritizing function $R \times I$. Functions R and I equal zero when the validation rule is satisfied.

by

$$f_{\mathsf{hb}}(y) = \begin{cases} \max\left\{\frac{y}{y^*}, \frac{y^*}{y}\right\} - 1 \text{ if } y > 0 \\ \mathsf{NA}, \text{ if } y = \mathsf{NA} \text{ or} y \leq 0. \end{cases} \tag{7}$$

where y^* is a predetermined reference value. A value y is considered invalid when $f_{\mathsf{hb}}(y)$ exceeds a threshold value h. The validation rule is therefore defined by $s = f_{\mathsf{hb}}$ and $V = [0, h]$. Note that the lower bound of V is a property of f_{fb}. Using the absolute difference as a distance function on the values of f_{hb}, $R(y, v)$ is easily derived:

$$R(y, v) \equiv \inf\{|f_{\mathsf{hb}}(y) - s'| : s' \leq h\} = \begin{cases} 0 \text{ if } f_{\mathsf{hb}}(y) \in [0, h] \\ |f_{\mathsf{hb}}(y) - h| \text{ if } f_{\mathsf{hb}} \geq h \\ \mathsf{NA} \text{ when } f_{\mathsf{hb}}(y) = \mathsf{NA}. \end{cases} \tag{8}$$

To compute the value for I, we need to express the range $V = [0, h]$ in terms of values of y. This can be done by finding the inverse of f_{hb} at $f_{\mathsf{fh}}(y) = h$ for the cases where $y \geq y^*$ and $y < y^*$. This then gives $y \in [y^*(h+1), (h+1)y^*]$ and we may write

$$I(y, v) \equiv \inf\left\{|y - y'| : y' \in [y^*/(h+1), (h+1)y^*]\right\} = \begin{cases} 0 \text{ if } y \in [y^*/(h+1), (h+1)y^*] \\ y^*/(h+1) - y \text{ if } 0 < y < y^*/(h+1) \\ y - (h+1)y^* \text{ if } y > (h+1)y^* \\ \mathsf{NA} \text{ if } y \leq 0 \text{ or } y = \mathsf{NA}. \end{cases} \tag{9}$$

Figure 3 gives a schematic overview of the behaviour of these functions. From the plot it is clear that although the impact I of invalid values on either side of the range is similar. Deviations on the low side of the allowed range are considered more severe (the severity function R falls off as y^{-1} on the range $(0, y^*/(h+1)]$ while on the same range I falls of as y). In the prioritizing function $R \times I$ this effect is dampened somewhat again.

21. Range edits can also occur for categorical variables, for instance when a variable, say *educational level* during data collection permits multiple values, e.g. {low, medium, high}. However, after selecting a certain subpopulation (say, persons teaching at a university) one may define the rule

$$\textit{educational level} = \text{high}. \tag{10}$$

Here, we have $s = \mathrm{Id}$ and $V = \{high\}$. To define an impact and severity function we need to define a distance function on the set of categories $\{\text{low}, \text{medium}, \text{high}\}$. Here, we take $\delta(y, y')$ which equals 1 if $y \neq y'$ and zero otherwise. We simply get

$$I(y, v) = R(y, v) = \delta\left(\textit{educational level}, \text{high}\right). \tag{11}$$

B. Multivariate, in-record validation

22. Multivariate validation functions restrict the combined value domain of multiple variables. Well-known examples are the sum rules appearing in business statistics. For example,

$$y_1 + y_2 = y_3. \tag{12}$$

Here, the score function s is defined by

$$s(y_1, y_2, y_3) = y_1 + y_2 - y_3, \tag{13}$$

when each of the values for all y_i are known and NA otherwise. The valid region is defined as $V = \{0\}$. More generally, each linear restriction on a numeric record $\boldsymbol{y}$ can be written as the combination of a linear score function $s_{\boldsymbol{a},b}(\boldsymbol{y}) = \boldsymbol{a}^\mathsf{T}\boldsymbol{y} - b$ and a valid region V. If $V = \{0\}$, the restriction is an equality. If V is an (open) half-line, the corresponding restriction is a (strict) inequality.

23. Using again the absolute difference between the computed score and the desired score as a measure of severity we get (when $\boldsymbol{a}^\mathsf{T}\boldsymbol{y}$ can be determined)

$$R(\boldsymbol{y}, \boldsymbol{a}, b) = \inf\{|s_{\boldsymbol{a},b}(\boldsymbol{y}) - u| : u \in V\} = \begin{cases} 0 \text{ if } \boldsymbol{a}^\mathsf{T}\boldsymbol{y} - b \in V \\ |\boldsymbol{a}^\mathsf{T}\boldsymbol{y} - b| \text{ if } \boldsymbol{a}^\mathsf{T}\boldsymbol{y} - b \notin V \\ \mathsf{NA} \text{ if } \boldsymbol{a}^\mathsf{T}\boldsymbol{y} = \mathsf{NA}. \end{cases} \tag{14}$$

which is readily computed for the cases of equality or inequality rules. Furthermore, if we choose the Euclidean distance to determine the impact function I, it is shown in van den Broek et al. [2014] that we may write

$$I(\boldsymbol{y}, \boldsymbol{a}, b) = \|\boldsymbol{a}\|^{-1} R(\boldsymbol{y}; \boldsymbol{a}, b). \tag{15}$$

24. The above example can be extended to include multiple, say k linear (in)equalities. Our validation rule is given by $\boldsymbol{A}\boldsymbol{y} \leq \boldsymbol{b}$, which may include equalities, strict inequalities ($<$) and inclusive inequalities ($\leq$). The score function is now a vector function $\boldsymbol{s}_{\boldsymbol{A},\boldsymbol{b}}(\boldsymbol{y}) = \boldsymbol{A}\boldsymbol{y} - \boldsymbol{b}$ and the valid region is a set of vectors given by $V = \{\boldsymbol{u} \in \mathbb{R}^k : \boldsymbol{u} \leq \boldsymbol{0}\}$. Using again the Euclidean distance to obtain a distance between records, we define the impact function on a numerical record as follows.

$$I(\boldsymbol{y}; \boldsymbol{A}, \boldsymbol{b}) = \inf\{\|\boldsymbol{y} - \boldsymbol{y}'\| : \boldsymbol{A}\boldsymbol{y}' - \boldsymbol{b} \leq \boldsymbol{0}\}. \tag{16}$$

Using an algorithm of Pannekoek and Zhang [2011], $\boldsymbol{y}'$ and therefore $I(\boldsymbol{y}; \boldsymbol{A}, \boldsymbol{b})$ can be determined. To compute the severity function R, we need to define a distance function on $\mathbb{R}^k$. If we choose the L_1 distance, we get

$$R(\boldsymbol{y}; \boldsymbol{A}, \boldsymbol{b}) = \inf\{|\boldsymbol{s}_{\boldsymbol{A},\boldsymbol{b}}(\boldsymbol{y}) - \boldsymbol{u}| : \boldsymbol{u} \leq \boldsymbol{0}\}, \text{ where } |\boldsymbol{u} - \boldsymbol{u}'| = \sum_i |u_i - u_i'|. \tag{17}$$

This value can be computed by row-wise application of Equation (14).

25. Equations (16) and (17) offer two rather natural ways to summarize the result of evaluating multiple similar validation rules over a dataset. In fact, we may treat $I(\boldsymbol{y}; \boldsymbol{A}, \boldsymbol{b})$ and $R(\boldsymbol{y}; \boldsymbol{A}, b)$ as new, single-valued validation rules by demanding for instance $I(\boldsymbol{y}, \boldsymbol{A}; \boldsymbol{b}) \leq h$. This allows one to introduce a certain amount of slack on a set of strict linear inequality rules without modifying the rule set itself.

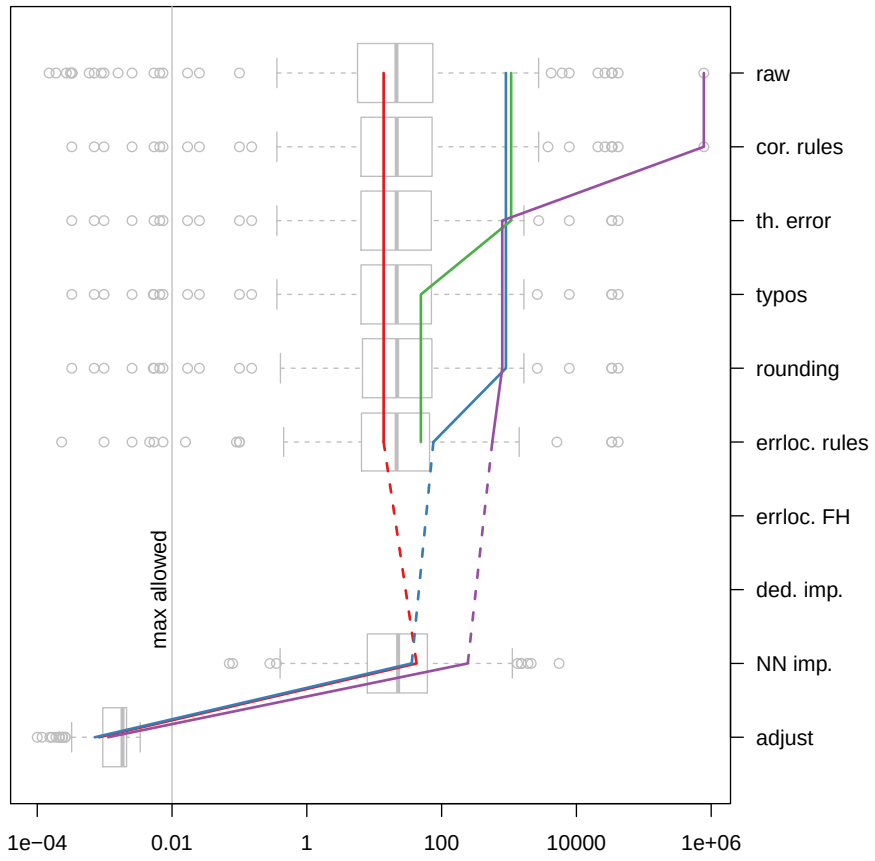

FIGURE 4. Impact function $I(\boldsymbol{y}; \boldsymbol{A}, \boldsymbol{b})$ [Eq. (16)] based on the Euclidean distance between actual records and the closest records satisfying a set of 78 linear (in)equality constraints as a function of processing step. Only non-zero distances (larger than 10^{-4}) are shown. The background boxplots show the distribution of the whole dataset (boxplot height is proportional to the number of records for which a distance was determined) while the lines indicate trajectories of a few individual records. A record is considered valid when $I(\boldsymbol{y}, v) \leq 0.01$.

26. As a demonstration, in Figure 4 we show the value of the impact function $I(\boldsymbol{y}; \boldsymbol{A}, \boldsymbol{b})$ of Equation (16) for 840 numeric records on child care institutions as a function of processing step. The data consists of some 48 variables which has to satisfy 78 (in)equality restrictions, including balance restrictions. The values were computed with the rspa package of van der Loo [2012]. A detailed overview of the methods applied to the dataset are given in Pannekoek et al. [2014] but briefly, records undergo 1) user-defined transformation rules, 2) thousand error correction, 3) correction of typing errors, 4) correction of rounding errors, 5) error localisation based on user-defined rules, 6) error localisation based on the principle of Fellegi and Holt, 7) deductive imputation of 8) nearest neighbour imputation and 10) minimal adjustment of imputed values. The boxplots in the background show the distribution of nonzero distances over the dataset. Since each point in the plot corresponds to a single record, we can plot traces of single records as they get processed. For example, the purple line on the far right corresponds to a record where one or more thousand errors are detected and fixed, next some values are removed by rule-based error localisation and after NN-imputation the imputed values are adjusted to obey the restrictions. The record represented by the red line shows that subsequent steps do not necessarily monotonously improve the quality: this records gets imputed with values that bring the

record further from the valid region before being corrected again. The record represented by the green line is treated fully by first repairing a typo, removing some values based on user-defined rules and deductively imputing new values. The blue line represents a record where all correction comes from error localisation, imputation and subsequent adjustment.

27. As another example, consider categorical data on *age class* and *marital status*. The data model is given by

$$\{\text{under-aged}, \text{adult}\} \times \{\text{married}, \text{never married}, \text{widowed}, \text{divorced}\}. \tag{18}$$

We introduce the restriction that 'an under-aged person cannot be or ever have been married'. The score function for the corresponding validation is the identity function. The valid region is most easily found by first defining the invalid region $\overline{V}$:

$$\overline{V} = \{\text{under-aged}\} \times \{\text{married}, \text{widowed}, \text{divorced}\} \tag{19}$$

and next compute V by taking the complement of $\overline{V}$.

$$\begin{aligned} V = \overline{\overline{V}} &= \{\text{adult}\} \times \{\text{married}, \text{never married}, \text{widowed}, \text{divorced}\} \\ &\cup \{\text{under-aged}\} \times \{\text{never married}\} \end{aligned} \tag{20}$$

Although the valid region has a more complex structure both the impact function and the severity function may be defined as 1 if a variable needs to be altered and 0 when a record obeys the rule. For multiple categorical rules, one may choose as a distance the minimum number of variables to alter so that the record can be made to obey all rules. In other words, this yields an error localisation problem for which several algorithms have been developed (see De Waal et al. [2011]).

28. A more complicated numerical example occurs when we define a conditional rule. For example, suppose we have a business survey with two types of personnel costs y_1 and y_2, and the number of staff working y_3. We define the restriction that 'if the total personnel cost is positive, the number of staff working must be positive as well'. The score function is a vector function defined as

$$s(y_1, y_2, y_3) = (y_1 + y_2, y_3). \tag{21}$$

The corresponding valid region V is again derived by realizing first that the invalid region $\overline{V} = (0, \infty) \times (-\infty, 0]$, so

$$V = \overline{\overline{V}} = (-\infty, 0] \times \mathbb{R} \cup (0, \infty) \times (0, \infty). \tag{22}$$

Here, V is a non-convex region in $\mathbb{R}^2$. Since it cannot be described with a single set of inequalities, the algorithm used to compute $I(\boldsymbol{y}; \boldsymbol{A}, \boldsymbol{b})$ cannot directly be applied. However, V can clearly be split up into two convex regions, and computing the distance to each region an then taking the minimum allows one to apply the algorithm anyway. It is clear however, that such an operation becomes quickly computationally expensive when multiple variables or multiple validation rules are treated.

IV. Summary and conclusions

29. In this paper we give an overview of our current research into methods that generalize the analyses of data validation results. As a step towards implementation, we pointed out relations between data validation activities and various international standards. Furthermore, we give a general model for a data validation action which allows us to derive three generic parameters: a boolean value stating whether data satisfies a certain quality requirement, an impact function that measures the effect on the data under validation and a severity function that measures the difference between a quality indicator value and its desired value. The latter two may be interpreted as a specific realisation of the 'discrepancy' and 'severity' values that are included in the VALS language. The main difference

is that where in VALS those values can be freely specified, in our model, these values follow naturally from the definition of a validation rule. Since these measures are defined regardless of the specific rule at hand, the question arises whether they can be generically implemented. We surveyed a number of often-used validation rules and determine the impact and severity function and find that for both linear record-wise restrictions and categorical restrictions, these measures permit a general and known algorithmic treatment, therefore allowing for a general implementation. It is an interesting question whether other classes of validation rules exist where such a general treatment is possible.

References

G. Brancato, R. Carbini, and Simeoni G. Metadata and quality indicators to report on editing and imputation to different users. In *Work session on statistical data editing*, UNECE conference of European statisticians, 2009.

Data Documentation Initiative. DDI, version 3.1, 2014. URL http://www.ddialliance.org/.

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.

ESSnet on SDMX2. Analyses of SDMX and DDI. Technical report, European Statistical System, 2013. URL http://cros-portal.eu/content/sdmxii-wp3-deliverables-28092012. last accessed 2014-03-06.

M.A. Hiridoglou and J.-M. Berthelot. Statistical editing and imputation for periodic business surveys. *Survey methodology*, 12(1):73–83, 1986.

J. Pannekoek and L.-C. Zhang. Partial (donor) imputation with adjustments. UNECE conference of European statisticians, 2011. URL Availableat:http://www.unece.org/stats/documents/2011.05.sde.html.

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. URL http://www.degruyter.com/view/j/jos.2013.29.issue-4/issue-files/jos.2013.29.issue-4.xml.

J. Pannekoek, M. van der Loo, and B. van den Broek. Implementation and evaluation of automatic editing. In *Work session on statistical data editing*. UNECE, 2014.

SDMX working group. SDMX, version 2.1, 2014. URL http://sdmx.org.

A. Simon. Validation syntax (VALS): Summary of the language. Technical report, Eurostat, 2013. URL https://webgate.ec.europa.eu/fpfis/mwikis/essvalidserv/index.php/Validation_in_the_ESS.

UNECE. GSBPM version 5.0, 2013a. URL http://www1.unece.org/stat/platform/display/GSBPM/GSBPM+v5.0.

UNECE. GSIM version 1.1, 2013b. URL http://www1.unece.org/stat/platform/display/metis/Generic+Statistical+Information+Model.

UNECE. CSPA version 1.0, 2013c. URL http://www1.unece.org/stat/platform/display/CSPA/Common+Statistical+Production+Architecture+Home.

B. van den Broek, M. van der Loo, and J. Pannekoek. Kwaliteitsmaten voor het datacorrectieproces. Technical Report 201408, Statistics Netherlands, 2014. (In Dutch).

Mark van der Loo. *rspa: Adapt numerical records to fit (in)equality restrictions with the Successive Projection Algorithm*, 2012. URL https://github.com/markvanderloo/rspa. R package version 0.1-4.