UNITED NATIONS
ECONOMIC COMMISSION FOR EUROPE

CONFERENCE OF EUROPEAN STATISTICIANS

Work Session on Statistical Data Editing
(Budapest, Hungary, 14-16 September 2015)

Topic (iii): Software tools and international collaboration

A FORMAL TYPOLOGY OF DATA VALIDATION FUNCTIONS

Prepared by Mark van der Loo, Statistics Netherlands

I. INTRODUCTION

1. Statistical data validation is an activity that pervades the statistical production process. Whether it is data collection, processing, estimation, or dissemination, most statistical offices will check data against desired properties at each step in the production sequence using formalized or informal processes. It is worth pointing out in this context, that the act of data validation (verifying whether data falls in a set of acceptable values) adds value to the data set under scrutiny. Namely, regardless of the outcome of a validation activity —a data set may either be flawless or have issues that limits its use, after a validation procedure we are more aware of its usefulness and we are able to better defend or refute its use for intended purposes. Validation exposes metadata that is implicitly present in the combination of data and the set of validation rules or procedures.

2. The topic of data validation has recently received increasing interest from authors and institutions within the European Statistical System (ESS). Examples are the typologies of Simón [2013a,b], the discussion on validation in the ESS by Henrard [2012] and the currently running ESSnet project on Validation [ESS, 2014]. Moreover, the SDMX consortium [2014] is currently developing a standardized Validation and Transformation Language (VTL), and the current author was involved in research that aims to analyze validation functions in a formal way [van der Loo and Pannekoek, 2014].

3. There are two recent working papers that aim at a classification of validation procedures. In Simón [2013a], validation levels are introduced that more or less mirror the statistical production chain where value is added to data step by step going from raw input to publishable data sets. In this paper, increasing levels of validation correspond to data satisfying checks against other data sources of increasingly broader origin (same or different file, source, provider, etc). A second paper of the same author [Simón, 2013b] gives an exhaustive overview and typology of validation rules used within Eurostat.

4. On one hand, an important merit of these papers is that they give a typology that is closely related to practical data validation procedures: comparing data between or within files, between or within data providers, and so on. On the other hand, efforts to standardize the communication and definition of validation procedures, including the ESSnet project and the development of VTL mentioned earlier, would benefit from a classification of validation activities that abstracts away from specificities of production processes, data (storage) types, sources and providers.

5. In this paper we document an attempt to study the concept of validation from a more formal point of view than we have usually encountered in the literature. By abstracting away from descriptions based on business processes, needs, or activities, we hope to uncover some structure that can be useful for expressing data validation rules (or procedures), for analyzing or comparing the capabilities of data validation languages or software, and for analyzing the outcome of data validation activities.

6. In the following we start by discussing the operational definition of validation used by the UNECE. Although we find the spirit of the definition is correct, we propose a reformulation such that a mathematical definition can be eventually be derived from it. To arrive at such a definition, and as a preparation for our classification, we discuss in Section III the process of measurement including the role of time and population dynamics. Having uncovered the central characteristics that identify a measurement (population at time of measurement, identity of the measured element and the measured variable), we can formally define a data point and accordingly a data set. In section IV we argue that a validation activity can be modeled as a function mapping a data set to the binary set $\{0, 1\}$ (or $\{0, 1, \mathsf{NA}\}$) and discuss some properties of such functions. In section V we propose to classify data validation functions according to the type of data set they validate. The identifiers for data points derived in section III play a central role here, and the fact that we find ten different classes depends on the fact that these identifiers are not entirely independent of each other. We show that the classification naturally leads to an ordering of validation classes in validation levels, based on the number of ways data points in a validated data sets may differ. Perhaps surprisingly, this ordering corresponds to the ordering commonly found in statistical value chains even though we made no assumptions about data processing methods. An overview of the classification is summarized in Figure 2. We discuss the validation levels and give examples of practical validation procedures for each type. We also discuss how our basic classification can be extended where relevant. Finally, we summarize discuss our findings and end with conclusions.

II. Definition of validation

7. At its heart, data validation is an attempt to falsify the assumption that values of a data set are acceptable as facts. Only after a sufficient amount of such attempts fail can data set be considered validated for a certain use. As is commonly the case in falsification, observed facts are checked against other observed facts for consistency. For example, one may check that someone's recorded age is greater than zero since we have never observed anyone with a negative age. Or, one may check the economic growth estimated from value added tax data against economic growth based on data from a business survey.

8. The operational definition in the UNECE glossary of terms on statistical data editing captures this idea by defining a set of acceptable values against which a single data field (data item) is compared:

> *An activity aimed at verifying whether the value of a data item comes from the given (finite or infinite) set of acceptable values.*

Indeed, the core of this definition (verifying data values against acceptable values) seems a useful operationalization of the concept of falsification. There are however, two issues with this formulation that prevent it from being formalized into a mathematical definition that allows a classification of validation activities.

9. The first issue is the fact that the UNECE definition is a teleological statement. A formal definition should be formulated such that given any object of phenomenon, one can check whether it is captured by the definition or not. Since the UNECE definition is a statement of intent, or purpose,

such a formalization is not possible. Simply put: one can not observe an activity and always verify with certainty whether it is performed with the intent to validate data or not.

10. The second issue is more practical. More often than not, data validation involves multiple data items (fields). Rather then validating a single data item, a validation activity aims to validate the combination of data items. Indeed, there are many involved data validation procedures that include extensive computations on whole data records, columns or other collections of data that yield a conclusion on the validity of the combination of data at hand.

11. Removing the sense of intent, and extending 'data item' to data set, we propose to reformulate the operational definition of validation as follows.

> *Validation is a procedure that verifies whether a collection of data falls in a set of acceptable values.*

The set of 'acceptable values' may be a set of possible values for a single field. But under this definition it may also be a set of valid value combinations for a record, column, or larger collection of data. We emphasize that the set of acceptable values does not need to be defined extensively. Rather often, a set of acceptable values is defined implicitly as the preimage of an involved calculation; for example when detecting outliers. Moreover, the set does not need to be fixed before the start of a validation procedure. For example, when an outlier is detected, one may find that the value is still acceptable, thereby conditionally expanding the range of allowed values for a certain collection of data.

12. Until now, we have not been very precise in defining what characterizes a data item or a collection of data. In fact, we have avoided the terms 'data set' and 'data point' until now, since in the next section we will carefully analyze the process of statistical measurement to arrive at a precise definition of these concepts.

III. Measurements and data

13. Before we continue with a formal definition of the concepts we need to classify data and measurements, let us look at the basic steps involved in obtaining a data point. Figure 1 shows a schematic timeline of such a process. At time t_u, a population element u is born. This may be a person, a company, a household or any other statistical object of interest such as a web site, a phone call, or a car crash. For the time being, we can think of u as a person. From the time of birth until its death our statistical object has one or more measurable properties X_t whose values may vary over time. For example, X_t may represent an income. At time τ, the element u is chosen for a measurement of income over the period $[t_x, t'_x)$. Now, this period may be in the past ('how much did you earn last year?'), in the present ('how much do you earn this year?') or even in the future ('how much do you expect to earn next year?'). The period $[t_x, t'_x)$ may be shrunk to an (approximated) point in time by letting $t'_x \to t_x$, for example by asking 'Of how many persons was your household composed at January 1 2015?'.

14. The above example exposes the key characteristics that locate a data point: we have a dynamical population, selection of elements from the current population and the moment of measurement. The period of time to which a measured value pertains is in itself not important for locating a data point since this can be considered metadata of the measured variable X_t. We stress that although we used a survey-like example, we emphasize that this view equally applies to administrative or big data sources. The main difference between these sources concern the nature of statistical objects and method of measurement but our discussion makes little or no assumptions about these concepts.

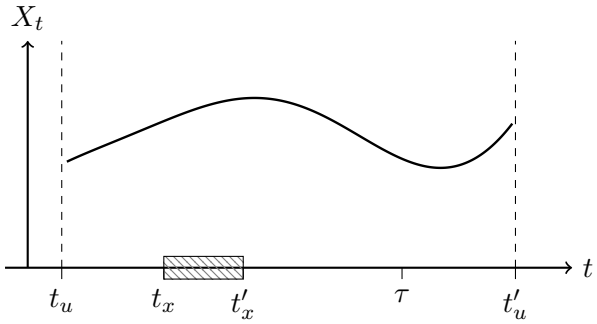

FIGURE 1. The various times involved in a measurement process. A population member u exists over the period $[t_u, t'_u)$. At the time of measurement τ a value of X_t is observed pertaining to the period $[t_x, t'_x)$. In principle, τ may be before, within, or after this period. Also, in stead of a period, one may choose a moment in time by letting $t_x \to t'_x$.

15. To formalize the above example, we follow the approach of Gelsema [2012], who derives a mathematical description of statistical information based on a description of real-world relations between statistical variables. Most notably for us, Gelsema suggests that statistical variables should be interpreted as mathematical functions that assign a value to an element of a statistical population – a definition which in fact corresponds to the definition of a random variable less the technical aspects imposed by measurability of probability spaces.

16. To capture the moment of measurement, we start by defining a finite universe $U = \{1, 2, \ldots, N\}$ whose elements represent every statistical object of a certain type that ever lived, lives now, or ever will live. For example, it may represent the set of all companies that ever existed, exist, and ever will exist, Similarly the set of all persons or households, but also web pages, e-mails or tweets, or events like phone calls or car accidents may be represented by such a set. The identity of statistical objects is fixed by their labeling in U. In this abstract view, and using T to denote the time line, a population can be written as a function

$$p : T \to 2^U,$$

where 2^U is the power set of U. Thus, $p(\tau)$ is the population at a certain time τ. Depending on the type (and definition) of object, the function p may or may not allow elements to appear, disappear, and reappear (resurrect) in $p(\tau)$. For example, bankrupt companies may be reinstalled but for persons the situation is different[1].

[1] Gelsema [2012] uses a slightly different formulation where the population is a subset of $U \times T$, i.e. each element of the population represents an object at a certain time. Regardless of subtle differences in how to identify statistical objects between the two approaches, the formulation used here is equivalent to Gelsema's in the sense that they do not alter our conclusions.

17. Given a population $p(\tau)$ a measurement can be done by first selecting an element, say i, from $p(\tau)$. We denote this as a function $s_{\tau,i} : 2^U \to U$, selecting element i from a subset $p(\tau)$ of U. Next, we can perform the measurement of a variable X_t, which is a function $X_t : U \to D$, where D is the domain of measurement. A complete measurement at time τ is now characterized by the following sequence of maps.

$$T \xrightarrow{p} 2^U \xrightarrow{s_{\tau,i}} U \xrightarrow{X_{t,\tau}} D. \tag{1}$$

To summarize: at some time τ, a population is fixed, an element selected and a measurement performed. Of course in practice, there will often be physical time between the determination of the target population (a certain copy of the population register for example) and the actual measurement. We can ignore this fact here, since even if the measurement is performed later, the statistical statements derived from it will still refer to the population at time τ. We therefore label both the selecting function and the measured variable with τ.

18. It is worth pointing out that the observed value of $X_{t,\tau}$ depends on the time of measurement as well as on its own natural evolution. For example, Zhang and Pritchard [2013] point out that values in administrative sources pertaining to some point or finite period in time may be updated several times afterwards. In short, one may read read $X_{t,\tau}$ as the observed value for X_t as measured at time τ. Since the dependence of variables on actual time t is of no relevance for the ensuing discussion, we will drop this label from now on.

19. The above discussion identifies the following aspects that need to be identified in order to localize a data point:

- the universe U;
- the time of measurement τ;
- the selected element i from $p(\tau)$;
- the measured variable $X_{\tau,j}$ where j labels the variable.

We now define a *data point* as a value, labeled with indices (U, τ, i, j). We will sometimes use less indices when for example U or τ are either fixed or unimportant. A *data set* is defined as a finite collection of data points. Examples of data structures that are covered by this definition include, a single data point, a single record (multiple variables, single object), a single column (multiple elements of U, single variable), a table, or a set of tables with records on different populations. In fact the set of all properly labeled data points available in a statistical office comprise a data set under this definition as well.

20. It is important to distinguish between the case where a measurement has not (yet) taken place and the case where a measurement has taken place but no value was obtained. To clarify this, consider a data set x, and a data retrieval function $f_x(U, \tau, i, j)$ that returns the value $x_{U,\tau,i,j} \in D_j$ if at time τ a measurement of $X_{\tau,j}$ took place for element $i \in p(\tau)$, and $\varnothing$ otherwise. If such a measurement did take place, it is still possible that the value is missing, because of nonresponse for example. In such a case f_x returns an NA code (not available). Hence, in many cases NA is an element of the measurement domain D_j.

IV. Validation functions

21. The purpose of a validation function is to decide whether a data set is fit for some predefined purpose. As stated in the definition of paragraph 11, this is done by verifying the set against a predetermined set of allowed values. If we denote with S the class of all possible datasets, then a

validation function v is a surjective function

$$v : S \twoheadrightarrow \{0, 1\}, \tag{2}$$

where 0 is interpreted as invalid (or 'false') and 1 as valid (or 'true'). We define v to be surjective on $\{0, 1\}$ since functions that always return 0 are contradictory (no dataset can be valid) and functions that are always 1 are non-informative (every possible dataset is valid). The valid region in S is now defined as the preimage $v^{-1}(1) = \{x \in S : v(x) = 1\}$. For a set of validation functions $\{v_i : i = 1, 2, \ldots, m\}$, the valid region is the intersection $\cap_{i=1}^{m} v_i^{-1}(1)$, expressing that all validation requirements must be satisfied.

22. The careful reader may note that validation procedures are usually fixed to a data point or data set of fixed dimension structure, which leaves one to wonder how a validation function can be defined on all data sets S. The answer is that we define that any data set that is not of the dimension structure that matches the input gets mapped to 1. So a validation function typically defines a large region of S as valid, simply because it does not attempt to falsify data in that region. Another interpretation is to say that most validation functions are partial functions of S, simply because they have to be constant over a large region of S.

23. In applications, the actual function definition is commonly user defined. Often such definitions take the form of simple rules but in principle, v can include arbitrary complicated computations, including aggregation, estimation or outlier detection. In the case of rules that are expressed as logical or comparison operators, we remind the reader that any such operator can be interpreted as a function. For example, the rule $y \geq z$ can be written as a function $\geq (y, z)$ taking a value in $\{0, 1\}$.

24. Since missing values (in the sense of NA described above) are a fact of life, it is in practice beneficial to propagate missingness when calculating a validation function and to extend the domain to $\{0, 1, \mathsf{NA}\}$. After all, the fact that a validation rule cannot be evaluated to 1 or 0 also conveys information about the quality of the data set. In fact, this definition is adopted by van der Loo and Pannekoek [2014] in a discussion on generic analyses of data validation outcomes, and the approach of NA propagation is common in statistical software. In our current discussion, the focus is on a classification of validation functions, based on the domain of such functions. For this purpose, we need not concern ourselves with the three-valued logic that comes with the definition of the extended output. Indeed, we adopt the view that the relation between the formal definition of formula (2) relates to the definition with the extended domain $\{0, 1, \mathsf{NA}\}$ much like how the set of real numbers relates to their representation as double precision numbers. While the real number line $\mathbb{R}$ is an important mathematical abstraction to reason with, it turns out that in practice it beneficial to work with a discrete representation of the set $(-\infty, -0] \cup [+0, \infty) \cup \mathsf{NaN}$ [IEEE, 2008]. Obviously, double precision arithmetic is inadequate when discussing the properties of real numbers but it is extremely important in applications. Similarly, validation functions are really aimed to make a decision about the data: valid or not valid. However, in practice it is very useful to allow missingness to propagate through a calculation, unless it is explicitly taken into account in the definition of the function.

25. We stress that in the definition of a validation function we do not distinguish between data to be validated and possible reference data that is used. Instead it is the combined set of input values that is validated. For example, it is not uncommon to check a value y_τ against an earlier value, say $y_{\tau-1}$ which is already deemed valid in an earlier process. One may for instance check whether $y_\tau / y_{\tau-1} \in [0.1, 10]$ or not. If this rule is not satisfied, drawing the conclusion that y_τ contains an error is a separate activity that is commonly called error localization.

26. Finally, and as an interesting aside, we note that since validation functions map to the set of Boolean values $\{0, 1\}$, the set of validation functions can be combined with Boolean operators under the following rules. Given validation functions v and w, we define $(v \wedge w)(x) = v(x) \wedge w(x)$, and similar for $\vee$ and $(\neg v)(x) = \neg v(x)$. The set of validation functions is not closed under boolean operations since it is possible to construct a v and w such that $(v \vee w)(x) = 1$ for all $x \in S$ (so $v \vee w$ is not surjective) or $(v \wedge w)(x) = 0$ for all $x \in S$. Obviously, the set of all validation functions is closed under negation. If we abandon the surjectivity demand, the set of validation functions is closed under Boolean operations. However, as discussed in paragraph 21, this comes at the price of allowing contradictory or noninformative validation functions.

V. Typology of validation functions

27. The definition of a validation function in Formula (2) suggest that we may classify validation functions by classifying the type of dataset a particular function works on. Since data validation is in essence based on comparing different sources of data, we propose here to classify validation functions based on the slices of S it works on. In particular, we will classify validation functions based on whether they are a function of a single or multiple instances of the indices (U, τ, i, j). But before continuing, let us look at a few examples.

28. Recall from Section III that we identify a data point by its universe U, the time of measurement t, the identity of the statistical object in the population at time of measurement $i \in p(\tau)$, and the measured variable $X_{\tau,j}$. Now consider the rule

$$y_{U,\tau,i,j} > 0. \tag{3}$$

This rule can be evaluated on a single data point. That is, one needs to fix a single U, a single time of measurement, a single statistical object i and a single variable j to test whether the rule is satisfied. The rule

$$y_{U,\tau,i,1} + y_{U,\tau,i,2} = y_{U,\tau,i,3}, \tag{4}$$

checks whether two variables from the same universe, time and same statistical object add up to a third. Thus, to evaluate this rule, one needs to fix a single U, a single time of measurement (τ), a single statistical object i and multiple variables j. Similarly, we could invent rules that use data on the same variable, statistical object and U, but measured at a different time.

29. The above examples illustrate that we may attempt a classification where validation functions validate over a single or multiple instances of one of the indices U, τ, i, and j. We may label each possible choice as a sequence of four labels s (single) and m (multiple). For example, the rule of Equation (3) can be classified $U\tau ij = ssss$ and the rule of Equation (4) can be classified $sssm$. The number of classes would be $4^2 = 16$ if the indices were completely independent. However, there are some restrictions. First of all, we note that although a statistical object can be represented in multiple universes (e.g. the universe of all households and the universe of households with more than two persons), it makes no sense to compare data where the only difference is that an object is selected by selecting from two (or more) different universes: we make the weak assumption that the selection process does not interfere with the rest of the measurement. We therefore exclude combinations of the form $U\tau ij = masb$ with $a, b \in \{m, s\}$. Secondly, once a universe is chosen, the measurable variables are defined as well, so we exclude combinations of the form $U\tau ij = mabs$ where a single variable would be defined on two universes [2]. Taking these restrictions into account we arrive at ten possible classes

[2] We define a variable as a characteristic of a statistical object. Two statistical objects of different types may have variables that may seem similar (e.g. one may speak of income for both persons and households) but since they are

Validation level				
0	1	2	3	4
ssss	sssm	ssmm	smmm	mmmm
	ssms	smsm	msmm	
	smss	smms		

FIGURE 2. Classification of validation functions, based on the combination of data being validated comes from a single (s) or multiple (m) universes U, times of measurement t, statistical objects i or variables j.

of input data, namely

$$ssss, sssm, ssms, ssmm, smss, smsm, smms, smmm, msmm, \text{ and } mmmm.$$

The above classes are both a complete and mutually exclusive characterisation of functions defined in Formula (2).

30. Observe that the number of m's occurring in a class label counts the number of ways in which data may vary while belonging to the validated data set. We therefore propose to use the number of m's as an indication of the extent, or level of validation. Grouping the validation classes per level yields the table of Figure (2). Going from lower to higher levels of validation in this model corresponds with the common practice where data is first tested against simple range checks (e.g. $y > 0$), and as data gets processed, more and more versatile data is used to verify the usability of a data set. This notion is somewhat remarkable since in our discussion we have not made any assumptions whether or not data is to be processed to yield statistical statements. In the following we will discuss each level with some examples.

A. Validation level 0

At this level we can only compare (functions of) a single data point with constants. Examples of rules in the class *ssss* are $y \in \{\text{male}, \text{female}\}$, $y > 0$, or $2y + 1 < 1$, where y is a single data point.

B. Validation level 1

31. At validation level 1, there are three options to extend the validated data set. Validation rules in the class *ssms* (multiple statistical objects) often compare functions of an individual element with a function of a column. An example is the rule: "the revenue of any single company may not exceed n times the median revenue over all observed companies from the same survey".

32. Rules of the class *sssm*, compare (functions of) different variables for the same statistical object within the same measurement. Well-known examples include the linear consistency checks used in business surveys (e.g. different sources of expense should add up to the total expense).

33. For validation functions of the class *smss* (multiple measurement times) the situation is a bit subtle. Although a different measurement time will often coincide with different times to which a measured value pertains, it is important to distinguish between the two. For example, suppose we measure the economic growth by surveying a panel of companies every month, asking them about the revenue of a month before. If we perform a check involving the ratio of revenues of the current measurement (pertaining to last month) and the previous measurement (pertaining to two months

functions of different populations they are not the same. In the language of probability theory, they are both random variables, but defined on different spaces Ω.

ago) for the same company, this check is classified as *smss*: a single universe (establishments), multiple measurement times, the same company and the same variable (last month's revenue). On the other hand, we may query a company once, asking for a series of revenue figures, one for each of the last n months. In that case, the same check is classified as *sssm*: a single universe with a single time of measurement on the same establishment, but asking for multiple variables of which two are used in the check.

C. Validation level 2

34. Here, we extend the dataset by varying two different characteristics. There are three options (three over two) since the universe is not varied. An example of a class *ssmm* rule (multiple statistical objects, multiple variables) is "*the total income of households observed in a survey must be larger than the total spendings observed in the same survey*". Informally, one may think of *ssmm* class validations as rules that compare aggregates of two columns. However, a rule that says "the income of a household cannot exceed n times the mean expenses over all households" also falls in this class. Another example would be a rule that compares the covariance between two variables from the same survey with a constant.

35. Rules of class *smsm* are multivariate functions, compared over different measurements. For example, suppose we measure income y and expenses x for company i at time $t-1$ and at time t. The rule $0.5 \leq |y_{\tau,i} - x_{\tau,i}|/|y_{\tau-1,i} - x_{\tau-1,i}| \leq 2$, states that we do not expect the ratio of the difference between income and expenses to vary with more than a factor of two between measurements.

36. A good example of *smms*-class rules are rules based on time series of aggregates of a single variable (where the observations underlying the aggregates are also measured at a different time). For example, suppose we measure the income of a (possibly dynamic) population of persons each year by a survey. Every year we get an (estimated) mean income $\bar{y}_\tau$. The rule $\bar{y}_\tau/\bar{y}_{\tau-1} > 1$ is an example of a rule in the class *smms*.

D. Validation level 3

37. There are two options to extend the validated dataset in three ways, either *smmm* (single universe, multiple measurement times, statistical objects and variables) or *msmm* (multiple universe, single measurement time, multiple statistical objects and variables). For an example of an *smmm* class rule, consider a survey that is repeated on a population and contains numerical variables X and Y. The rule stating that the covariance between X and Y may not vary more than say, 10% between surveys is a rule in this class: it compares multivariate aggregates over different measurements.

38. Examples of validation rules of class *msmm* involve measurement at the same time of a two different universes (populations). A practical example would be a case where one fixes the state of two administrative sources for different populations at a single time. For example, we could take phone call records for a certain day (from the universe of phone call records) and the population register of a country at the same day. An example of a *msmm* class rule is one where we compare the total number of minutes called per city with the number of inhabitants, demanding that the ratio between them is between certain limits. Essential to rules involving multiple universes is that there is a reason to suspect an interaction or correlation between the behaviour of two different populations.

E. Validation level 4

39. In this class we validate a combination of data that involves measurement of different populations over different times of measurement. As the class involves a very broad range of data, it is harder to find correlations that serve as a rationale for validation rules. However, one example would be the same as the example above (phone records and population register) except that the phone records are collected on a different date then the population register.

F. Extending the typology

40. In this work, we have limited the parameters that localize a data point to a few very basic characteristics: the universe representing statistical objects (U), the time of measurement (τ), the chosen object to investigate (i) and the variable measured j. We have not dealt with cases where modes of measurement are used for the same variable at the same time and population — a topic which is currently of great interest in the field of (household) survey strategies. Indeed, we have attempted to set up a classification that is independent of such choices by analyzing the pure process of measurement. However, such cases can be included by either adding extra characteristics to the list $U\tau ij$ if this is relevant for the classification of validation functions. Alternatively one can take the point of view that a different measurement mode defines a different variable, in which case the current classification is sufficient.

VI. Summary and conclusions

41. Data validation is an essential part of any statistical production chain that in fact creates value by itself. In this paper we have given an operational definition that can be formalized in terms of a mathematical function. By carefully studying the process of measurement, we derive a few characteristics that are minimally necessary to identify a data point and hence a data set. By classifying the type of domains of which a validation procedure can be a function, we classify validation procedures into ten different classes. By counting the number of ways data characteristics may vary over the domain of validation functions, we arrive in a natural way at a definition of validation levels. For future work, it will be interesting to investigate how for example the generic analyses of validation functions of van der Loo and Pannekoek [2014] works out for each of the classes defined here. Moreover, it will be interesting to see how the validation levels defined here work out when applied to practical cases, for example when comparing software.

VII. Acknowledgements

The author likes to thank Tjalling Gelsema, Dick Windmeijer, Jeroen Pannekoek and Olav ten Bosch for carefully reading the manuscript and fruitful discussions on this topic.

References

ESS. Quality assurance framework for the European statistical system. Technical report, European Statistical System, 2012. URL http://ec.europa.eu/eurostat/documents/64157/4392716/qaf_2012-en.pdf/.

ESS. ESSnet on validation, 2014. URL http://www.cros-portal.eu/content/validat-foundation.

T. Gelsema. The organization of information in a statistical office. *Journal of Official Statistics*, 28 (3):413–440, 2012.

M. Henrard. Proposal of a revised approach for data validation within the european statistical system. In *United Nations Economic Comission for Europe Work Session on Statistical Data Editing*, Oslo, 2012. URL http://www.unece.org/fileadmin/DAM/stats/documents/ece/ces/ge.44/2012/13_Eurostat.pdf.

IEEE. *Standard for floating point arithmethic*, 2008. IEEE Std. 754-2008.

SDMX consortium. Validation and transformation language rfc 1.0, September 2014. URL http://sdmx.org/?p=1957.

A. Simón. Definition of validation levels and other related concepts. Technical report, Eurostat, 2013a. URL https://webgate.ec.europa.eu/fpfis/mwikis/essvalidserv/images/3/30/Eurostat_-_definition_validation_levels_and_other_related_concepts_v01307.doc.

A. Simón. Exhaustive and detailed typology of validation rules. Technical report, Eurostat, 2013b.

UNECE. *Glossary of terms on statistical data editing*, 2000. United Nations Economic Commission for Europe, United Nations. URL http://ec.europa.eu/eurostat/ramon/coded_files/UN_Glossary_terms_stat.pdf.

M. van der Loo and J. Pannekoek. Towards generic analyses of data validation functions. In *United Nations Economic Comission for Europe Work Session on Statistical Data Editing*, Paris, 2014. URL http://www.unece.org/stats/documents/2014.04.sde.html.

L.-C Zhang and A. Pritchard. The integration of survey and administrative data. In *3rd European Establishment Statistics Network, Nuremberg, Germany*. ENBES, 2013. URL http://enbes.wikispaces.com/Zhang+et+al+-+The+integration+of+survey+and+administrative+data.