**What is Data Editing?**

Data editing is the activity aimed at detecting and correcting errors in data.

The editing procedure usually includes three phases:

* the definition of a consistent system of requirements (checking rules),
* their verification on given data set (data validation or data checking) and
* investigation and correction of the errors, elimination or substitution of data which are in contradiction with the defined requirements (data imputation).

This schema shows the relationship between three concepts widely utilised in the context of data quality:

* Establishment of checking rules;
* Detection of outliers or potential errors;
* Communication of the detailed problems to the "actors" in the best position to investigate them;
* Corrections of the errors based on appropriate investigations or automatic imputation;
* Technical activities, e.g. data analysis, which are not part of the agreed set of checking rules;
* Activities of other nature, for instance compliance monitoring which is a set of governance processes that are meant to stimulate EU Member States to respect their obligation to apply systematically the EU statistical legislation.

**What is Data Validity?**

An activity aimed at verifying whether the value of a data item comes from the given (finite orinfinite) 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.

Data validation assesses the plausibility of data: a positive outcome will not guarantee that the data is correct, but a negative outcome will guarantee that the data is incorrect.

**Data validation procedure**

Data validation is a decisional procedure ending with an acceptance or refusal of data as acceptable. The decisional procedure is generally based on rules expressing the acceptable combinations of values. Rules are applied to data. If data satisfy the rules, which means that the combination expressed by the rules is not violated, data are considered valid for the final use they are intended to.

Sometimes the rules used in a validation procedure are split in hard/fatal edits and soft/query edits and the not acceptable values are classified either as ‘erroneous’ or ‘suspicious’ depending on whether they fail hard edits or soft edits. Hard edits are generally rules that must necessarily be satisfied for logical or mathematical reasons (e.g., children cannot be older than their parents). An example of query edits on statistical data editing is “a value that, compared to historical data, seems suspiciously high” while for fatal edits is “a geographic code for a Country province that does not exist in a table of acceptable geographic codes”.

In addition to this information, a data validation procedure may assign a degree of failure (severity). Taking the example previously mentioned for soft edits, the severity can be evaluated by measuring the distance of the actual values to the expected values (e.g. based on historical data).

[TODO] Metrics section

The data validation process is an iterative procedure based on the tuning of rules that will converge to a set of rules that are considered the minimal set of relations that must be necessarily satisfied.

When the validation fails, it may produce three types of error (the severity):

* Fatal error: the data are rejected;
* Warning: the data can be accepted, with some corrections or explanations from the data provider;
* Information: the data are accepted.

**Why data validation - Relationship between validation and quality**

The purpose of data validation is to ensure a certain level of quality of the final data.

Nevertheless, quality has different dimensions in official statistics: relevance, accuracy, timeliness and punctuality, accessibility and clarity, comparability, coherence, completeness. Hence, it is important to establish on which components data validation is concerned with.

Data validation focuses on the quality dimensions related to the ‘structure and content of the data’, that are accuracy, comparability, coherence.

**Accuracy**

Data accuracy refers to whether data values are correct. To be correct a data value must be both the right value and be represented in an unambiguous form, two characteristics of data accuracy are form and content.

Form

“Form is important because it eliminates ambiguities about the content,”. Form dictates how a data value is represented, For example P-Codes of a country (Yemen) could be recorded as ‘0021’ but to remove the ambiguity the full P-Code should be used with country name prefix i.e. YE in this case to make P-Code ‘YE0021’

Content

As for content, “two data values can be both correct and unambiguous yet still cause problems.” This is a common challenge with free-form text, such as a city name. “The data values ST Louis and Saint Louis may both refer to the same city, but the recordings are inconsistent, and thus at least one of them is inaccurate.” Consistency is a part of accuracy, because “inconsistent values cannot be accurately aggregated and compared. Since much of data usage involves comparisons and aggregations, inconsistencies create an opportunity for the inaccurate usage of data.”

Validity versus Accuracy

Defining all values that are valid for a data element is useful because it allows invalid values to be easily spotted and rejected from the database. However, we often mistakenly think values are accurate because they are valid. For example, if a data element is used to store the name of a state of a country. The value of ‘Education’ would be invalid but the value of New York would be valid but inaccurate if the data is being collected in Yemen.

**Can 100% Data Accuracy be achieved?**

The short answer is no, You can get accurate data to a degree that makes it highly useful for all intended requirements.

**Coherence and comparability**

The general definition of coherence and comparability claims that statistics should be consistent internally, over time and comparable between regions and countries.

Coherence and comparability aspects are definitely important for the data validation process. Validation rules and the process of confronting the data set with validation rules, the process of detecting errors and flagging them should be coherent and consistent internally and between countries, based on common standards with respect to the scope and national deviations.

**Clarity and accessibility**

‘Accessibility and clarity’ is a quality dimension checked in a data validation process, it is related to the IT formal checks (Structural Validation) that are performed in order to be able to read without any misunderstanding the data.

**Timeliness**

Timeliness is not a quality dimension checked by a validation procedure. Nevertheless, it is important to remark that it has a strong connection with a data validation procedure. Timeliness can be seen as a constraint when designing a data validation procedure. For instance, in case of complex checks and time demanding editing procedures, a less restrictive data validation process allowing a higher amount of errors in data may be designed to meet the expected timing of the release of final data.

A final remark about the concept that data validation “aims at verifying whether data have a certain level of quality”: It is indeed true that data validation cannot ‘ensure’ a level of quality. What it can more realistically provide is that at least a certain level of data consistency considered as the minimum requirement for having acceptable data, is reached. This results not in perfect data, but in ‘plausible’ data.

**How to perform data validation: validation levels and validation rules**

Because of the variety of validation steps and procedures and because of the way validation procedures pervade statistical production processes, it is desirable to be able to judge to what extent a data set has been validated (validation level) by validation procedures applied to it. Moreover, as statistical processes age and mature, the number of validation procedures and rules tend to grow organically, generating a need for maintenance. Finally, one would like to be able to compare statistical processes and statistical software in view of their abilities to validate data.

Clearly, the above tasks would be easier, if there was some sort of system that classifies validation levels, validation rules and procedures into disjoint subtypes.

To develop any classification system, one needs to consider what principle or principles separate the different classes. For validation rules and procedures, the following come to mind or have been encountered in literature:

* automated versus manual;
* objective versus subjective/expert opinion;
* structural validation versus content validation;
* set being validated: in-field, in-record, cross-record, cross-data set, etc.;
* place in the statistical process: input, throughput, output;
* type of validation rule: equality, inequality, logical rule,…

and sure there are many more options. Depending on the task at hand, different classifications may be useful, as long as they are both exhaustive and mutually disjoint.

It is generally assumed that there are basically two general categories:

1. Technical integrity of the file, i.e., consistency with the expected IT structural requirements (Structural Validation)
2. Logical and statistical consistency of the data (Content Validation)

The second category is generally split into different sub-categories (levels) involving more and more information. The two general categories can then be expanded forming the following validation levels from a business perspective.

* Validation level 0: consistency with the expected IT structural requirements
* Validation level 1: consistency within the data set
* Validation level 2: consistency with other data sets within the same domain and within the same data source
* Validation level 3: consistency within the same domain between different data sources
* Validation level 4: consistency between separate domains in the same data provider
* Validation level 5: consistency with data of other data providers

**Validation level 0: consistency with the expected IT structural requirements**

At this level, it is checked the consistency of the data with their expected IT requirements, for instance

* if the file has been sent/prepared by the authorised authority (data sender);
* if the column separator, the end of record symbol are correctly used;
* if the file has the expected number of columns (agreed format of the file);
* if the column have the expected format of the data (i.e., alphanumeric, numeric, etc.)

For these quality checks only the structure of the file or the format of the variables are necessary as input.

**Validation level 1: consistency within the data set**

It is checked the consistency within the elements of the data set. For these quality checks, it is needed only the (statistical) information included in the file itself.

For instance:

* check whether the number included in column 4 is not negative (as expected);
* check whether the year in the second column is 2011, as in the file name;
* check whether the content of the third column is one of the codes of the dictionary "Sex";
* check whether the content of the first column is consistent with the data sender (let's assume that there is a dictionary including the list of the data senders associated to the specific data set): data for Luxembourg should not be sent by another country.
* based on information available before data collection (for example from previous survey or other sources) one could establish a "plausibility range" for a certain variable (for instance number of components of a household).
* check consistency at (micro-level) of two (or more) variables: a certain combination of codes is illogical, a variable has to be reported only for a certain combination of codes.
* check consistency at macro-level of two (or more) variables: Total inhabitants = male inhabitants + female inhabitants, or Female inhabitants = (total inhabitants / 2) +/- 10%

**Validation level 2: consistency with other data sets within the same domain and within the same data source**

Validation levels 2 is concerned with the check of consistency based on the comparison of the content of the file with the content of "other files" referring to the same statistical system (or domain) and the same data source.

For instance:

* Case a) the "other files" can be other versions of exactly the same file. In this case the quality checks are meant to detect "revisions" compared to previously sent data. Detection and analysis of revisions can be useful for example to verify if revisions are consistent with outliers detected in previous quality checks (corrections) or to have an estimate of the impact of the revisions in the "to be published" results, for the benefit of the users.
* Case b) the "other files" can be versions of the same data set referring to other time periods. These checks are usually referred to as "time series checks" and are meant to verify the plausibility of the time series.
* Case c) the "other files" can refer to other data sets from the same data provider (e.g., Countries in the ESS), referring to the same or other correlated time periods. Sometimes a group of data sets (same country, same reference period) is sent at the same time.

Example: three files could be sent at the same time, from the same country and referring to the same time period: one file includes data for "females", one for "male" and one for "total". Consistency between the results of the three files can be checked.

Another example: results from annual data sets can be compared with the results of the corresponding quarterly data sets.

**Validation level 3: consistency within the same domain between different data sources**

Validation levels 3 is concerned with the check of consistency based on the comparison of the content of the file with the content of "other files" referring to a different data provider on the same harmonized statistical system or domain (sharing common standards with respect to scope, definitions, units and classifications in the different surveys and sources).

For instance:

Case d) the "other files" can refer to the same data set, but from another data provider (e.g., Countries of the ESS, sharing harmonized methodologies). Mirror checks are included in this class. “Mirror statistics involve coherence, geographical comparability as well as accuracy issues” (Eurostat, 2014). Often such statistics is important for data analysis on Eurostat level. Mirror checks verify the consistency between declarations from different sources referring to the same phenomenon, e.g., export declared by country A to country B should be the same as import declared by country B from country A.

**Validation level 4: consistency between separate domains in the same data provider**

Validation level 4 could be defined as plausibility or consistency checks between separate domains available in the same Institution. The availability implies a certain level of "control" over the methodologies by the concerned Institution.

These checks could be based on the plausibility of results describing the "same" phenomenon from different statistical domains. Examples: unemployment from registers and from Labour Force Survey, or inhabitation of a dwelling (from survey of owners of houses and dwellings vs. from population register)

Checks could also be made between results from correlated micro-data and macro-data sources.

Other plausibility checks could be based on known correlations between different phenomena: for example external trade and international transport activity in ports.

**Validation level 5: consistency with data of other data providers**

Validation level 5 could be defined as plausibility or consistency checks between the data available in the data provider (Institution) and the data / information available outside the data provider (Institution). This implies no "control" over the methodology on the basis of which the external data are collected, and sometimes a limited knowledge of it.

To summarize, the classification of validation levels presented above implicitly assumes a growing degree of complexity from one level to another. However, this must not necessarily be reflected by a growing technical complexity of the validation checks themselves. From the technical point of view, the distinction made with respect to data sets is an artifice, since data sets and files could be merged into single databases in advance of implementing the checks.

A likely rise in complexity might be regarding organizational and management matters. On the higher levels of this classification more parties and stake-holders will be involved, potentially with different needs and requirements regarding data validity. This certainly tends to make it more difficult to harmonize technical and methodological concepts. However, this may depend very much on the concrete situation and circumstances.

**Validation rules**

The validation levels, as anticipated in the examples of validation levels, are verified by means of rules. Rules are applied to data, a failure of the rule implies that the corresponding validation level is not attained by the data at hand.

As explained in the beginning of section 4, a first broad classification of validation rules distinguishes rules to ensure technical integrity of the data file (Category A “Structural Validation”) and rules for logical/statistical consistency validation (Category B “Content Validation”). The distinction is useful since the rules used in the two contexts can be very different. Examples for the different rule types have been reported by the respondents of the ESSnet ValiDat Foundation survey. Some of them will be presented further below:

**Structural Validation: Rules to ensure technical integrity of a data file format and structure:**

* formal validity of entries (valid data type, field length, characters, numerical range)
* presence of an entry
* no duplicate units
* all the values in a field of one data set are contained in a field of another data set (for instance contained in a code list(s)
* each record has a valid number of related records (in a hierarchical file structure)

**Content Validation: Rules for logical validation and consistency could be classified using the two typology dimensions presented in table 1, e.g. identity vs. range checks (1) and simple vs. complex checks.**

Table1: Categories of a 2-way typology for validation rules for logical validation and consistency

|  |  |  |
| --- | --- | --- |
| **Typology** | **Types of checks** | |
| 1 | Identity checks | Range checks   * bounds fixed * bounds depending on entries in other fields |
| 2 | Simple checks, based directly on the entry of a target field | More “complex” checks, combining more than one field by functions (like sums, differences, ratios) |

Also, rules are often implemented as conditional checks, i.e. they are only checked, if a certain condition holds. This can be regarded as another property of a rule and might be considered as additional “dimension” of the rule typologies (for both rule sets, Categories A and B).

Typical conditions of a conditional check mentioned by the ESSnet ValiDat Foundation survey respondents are

* if “age under 15” (then marital status must be not married), or
* if “legal form: Self-Employed” (then number of self-employed" must exceed 0), or
* if “status in employment = compulsory military service” (then sex must be male), or
* if “no. of employees not zero” (then wages and salaries must be greater than zero), or
* if “enterprise reports production of goods” (then it should also report costs for raw material), etc.

Of course there might be several conditions combined by logical AND or OR statements.

Table 2 below presents at least one example2 for each rule type in Category A.

For the rule types of Category B, table 3 provides examples2.

Table 2: Examples of rules to ensure technical integrity of a data file (Structural Validation)

|  |  |
| --- | --- |
| Formal Validity of   * Data type * Field Length * Characters * Numerical range | * Telephone Number: Numeric * *Date:* If Date is given as text it should be 8 characters long * Date: If Date is given as text it should contain only numbers. * Month: Month of arrival in the country must be in {1,...,12} |
| Presence of an entry | * Persons in households: It is checked whether all have responded. * Code for Sex: no missing data. |
| No duplicate units | *Holding ID*: Each holding has a unique ID number, duplicate ID numbers are not allowed within the data set |
| All the values in a field of one data set are contained in a field of another data set (for instance contained in a codelist) “code list check” | Occupation: Field “Occupation” must contain only entries from a list of valid ISCO-08(COM) codes at 3 digits level  Country of origin: Field "country of origin" must contain only entries from a list of valid ISO country codes |
| Each record has a valid number of related records (in a hierarchical file structure)  “Cardinality check” | Number of members of a family: the aggregated number of persons in each family must be equal to the number of individual rows in the data set corresponding to the members of that family |

Table 3: Examples of rules for logical validation and consistency (Content Validation)

|  |  |  |
| --- | --- | --- |
| Identity checks | In a sheep survey : “Milk production” must be equal to “milk disposal” | Employment: “Number of persons engaged” must be equal to the sum of “employees” and “self-employed persons” |
| Range checks  - bounds fixed | Working hours (monthly): “Hours worked” must be between 0 and 168 | Average poultry weight:  “weight of poultry” divided by “number of poultry” must be between 0.03 and 40 |
| - bounds depending on entries in other fields | Cereal production: “Organic cereals” must be less or equal to “cereals” | External services:  “Expenses on external services” must be greater or equal to  “payment for agency workers” plus “telecommunications” plus “business trips of company personnel” |

Of course, we should take into consideration that some types could be inherent to both categories, e.g. presence of an entry, etc. More information on exhaustive typology of validation rules for statistical purposes could be found in the Eurostat document “Main types of data validation rules”.

Notably, not all cross-combinations in the 2-way representation of rule types used to define the fields in table 3 are “necessary” from a language perspective. For example, any range check of the type “complex” can be expressed as range check with fixed bounds. For illustration, consider the instance provided in table 3 for checking expenses on external services. This rule would be equal to the following rule with a fixed bound of zero:

“Expenses on external services” minus “payment for agency workers” minus “telecommunications” minus “business trips of company personnel” must be greater or equal to zero.

Moreover, any check of the type “complex” might be implemented as well as check of the type “simple”: According to our definition, a “complex” check is a check combining more than one field by functions (like sums, differences, ratios). Of course one might implement into the procedures of a statistic a step which derives new variables implementing such “functions”. If the validation step is carried out after such a preparatory step, all “complex” checks will be “simple”. This has also been reported as actual practice by one of the survey respondents: “Variables that are a combination of elementary ones are always implemented by combining elementary variables in the output process”.

An extended list of validation rules is provided in Appendix A.

Functions

GSBPM. Analyse phase, sub-process 6.2. Validate outputs

The last sub-process is the 6.2 (‘Validate outputs’).

“This sub-process is where statisticians validate the quality of the outputs produced, in accordance with a general quality framework and with expectations. This sub-process also includes activities involved with the gathering of intelligence, with the cumulative effect of building up a body of knowledge about a specific statistical domain. This knowledge is then applied to the current collection, in the current environment, to identify any divergence from expectations and to allow informed analyses. Validation activities can include:

* checking that the population coverage and response rates are as required;
* comparing the statistics with previous cycles (if applicable);
* checking that the associated metadata and paradata (process metadata) are present and in line with expectations
* confronting the statistics against other relevant data (both internal and external);
* investigating inconsistencies in the statistics;
* performing macro editing;
* validating the statistics against expectations and domain intelligence”

The checks that are not usually considered as a part of a ‘data validation’ procedure (i.e., the first and the third item

Remark. The attention of this validation step is on the output of the ‘process’ step. It means that data are already processed, e.g., statistical data editing and imputations are done.

[What component we are speaking about here]

According to these definitions, data validation can be interpreted as a business function corresponding to different business processes, which means that data validation can be performed at different stages of the production chain, in fact data validation refers to different phases of GSBPM. These phases, composed of process steps, are distinguished by their process inputs, i.e., any instance of the information objects supplied to a Process Step Instance at the time its execution is initiated.

Data validation is in general described by an application of rules to a set of data to see whether data satisfies the rules. Data which satisfies the rules are considered validated. The output of the process is a data set with indicators addressing which data are considered acceptable and hence validated, indicators of metrics, ..., and indicators measuring the severity of the failure (if any).

The data sets are to be considered in a broad way, they can be composed of microdata or aggregates and they can have a longitudinal part or not.

GSIM defines data set as:

“A Data Set has Data Points. A Data Point is placeholder (for example, an empty cell in a table) in a Data Set for a Datum. The Datum is the value that populates that placeholder (for example, an item of factual information obtained by measurement or created by a production process). A Data Structure describes the structure of a Data Set by means of Data Structure Components (Identifier Components, Measure Components and Attribute Components). These are all Represented Variables with specific roles.

Data Sets come in different forms, for example as Administrative Registers, Time Series, Panel Data, or Survey Data, just to name a few. The type of a Data Set determines the set of specific attributes to be defined, the type of Data Structure required (Unit Data Structure or Dimensional Data Structure), and the methods applicable to the data.”

**The data validation process life cycle**

In order to improve the performance of a statistical production process by managing and optimizing the data validation process, it is useful to describe the data validation process life cycle.

First, the process should be seen as a dynamic and complex process. Adapting validation rules may influence not only in the scope of one data set or one statistical domain, but also to all statistical domains. For instance, the optimization of efficacy and efficiency of the validation rules should take into account their assessment in the previous occasion, relations of indicators, etc. Second, the process should be seen as an integral part of the whole statistical information production process.

The data validation life cycle involves the activities directly linked to each statistical domain for the definition and execution of data validation. This cycle starts by designing the data validation process for the statistical domain or inter statistical domain, with an overall study of the data sets, variables and its relations to find a list of suitable and effective validation rules. In the implementation phase, these validation rules are described in common syntax, formalized, tested and refined, discussed and evaluated by stakeholders. During the execution phase, data are checked against the rules; validation results are measured and quantified. These outputs will be reviewed to improve the list of validation rules.

The data validation life cycle process includes the review of obtained statistical data through data editing, in fact the output of this task is used to improve the data validation procedure in an iterative way.

Data validation process is an integral part of the whole statistical information production process. Validation tasks and controls are performed by several stakeholders with a wide range of responsibilities. The data validation process life cycle should provide clear and coherent allocation of actions and responsibilities to ensure the highest performance, while reducing the possibility of mistakes. Though, allocation of responsibilities for each phase of data validation life cycle is hardly possible due to the complexity of data validation procedure and because this is strongly related to the particular organization structure.

Designing validation rules and rule sets for a data set implies the distribution of validation tasks in the statistical production chain to be proposed to the decision making structures. This distribution of responsibilities should be designed following the principle of "the sooner the better" as it is commonly agreed that the cost in terms of resources, time and quality of fixing data errors is lower as closer it is to the data source.

Figure 6: Data validation process life cycle



**Design phase**

The design of a data validation process is a part of the design of the whole survey process. The data validation process has to be designed and executed in a way that allows for control of the process. The design of the validation process for a data set in or between the statistical domains requires setting up the validation rules to be applied to the data set.

These set of validation rules should be complete, coherent, and efficient and should not contain any inconsistencies. Designing a set of validation rules is a dynamic process. Validation rules should be designed in collaboration with subject matter specialists and should be based on analysis of previous surveys. Consistency and non-redundancy of rules should be verified. Validation rules should be designed cautiously in order to avoid over-editing. Effective validation rules can be obtained by differently combining approaches and “best practices”.

In this phase the validation process should be planned and documented for further progress monitoring. The overall management of the process and the interfaces with the other sub-processes should be considered. For each phase the resources and time needed to implement, test, execute, review and document should be planned.

This is the phase where survey designers, questionnaire designers, validation and editing specialists and subject matter experts have to co-operate.

Activity descriptions

* Assess quality requirements for data sets
* Overall study of data sets, variables and their relations
* Determine satisfactory set of validation rules for the data. In order to make data production process more efficient, reducing time and human resources, but considering quality requirements.
* Assess responsibilities and roles. Document who is doing what; who is responsible for different actions; who is accepting and adopting the validation rules, etc.
* Integrate the data validation process in the overall statistical production process. Design the connections with other phases of the statistical production processes.
* Improvement of the validation according to the results of the review phase

A document with the form of guidelines with some theoretical background, examples and best practices could support the task of the domain manager when designing the entire validation process.

**Implementation phase**

Once the data validation process has been designed, it has to be implemented with a parameterization, thoroughly tested, tuned and become productive.

The validation process should be tested before it is applied. Validation rules and editing techniques and methods should be tested separately and together. It is important to realize that once the validation process is implemented in the actual survey process, only slight changes should be made to monitoring and tuning in order to avoid structural changes.

Common definitions and descriptions applied to data validation are required for a common understanding of the whole validation process.

A proper documentation of the validation process is an integral part of the metadata to be published. The aim of documentation is to inform users, survey managers, respondents, validation and editing specialists about the data quality, the performance of the process, its design and adopted strategy. The documents can be of three types: methodological, reporting and archiving.

The validation rules should be written in an unambiguous syntax that could allow communicating the rules amongst the different actors in the production chain and could also be interpreted by IT systems.

People working on validation and related aspects should have a sound knowledge of the methods that can be adopted, aware about the links between the validation and the other parts of the statistical production process. At this phase cooperation from methodologist and IT specialist should be very concise.

Activity descriptions

* Validation rules are formalized and described in a common syntax.
* Determine metrics for data validation rules, assessment of validation process and validation rules. Validation rules should be assessed for quality (clear, unambiguous and consistent, saving time resources).
* Testing. Apply validation rules to test data (real data, artificial data) and producing indicators.
* Test results (indicators, validation rules, metrics, quality aspects, etc.) are evaluated by stakeholders (Eurostat, Member states, Domain managers, etc.). Reporting documents on test results and evaluation should be prepared and saved for review phase.
* Refinement of validation rules according to the test results and consultations with stakeholders
* Documenting. Data validation rules should be well documented – documents depend on the purpose and the final user: producers, users of the results, survey managers or methodologists.

**Execution phase**

The execution phase consists of identifying values that are not acceptable with respect to rules expressing logical, mathematical or statistical relationships. This process usually consists of a set of integrated validation methods dealing with different type of errors. This allows assessing the quality of the data and helps to identify error sources for future improvements of statistical production process.

The result of execution phase is a flag indicating acceptable and not acceptable data, and generally a score measuring the degree of severity of failure.

A standard communication of error/warning messages may increase the global efficiency of statistical production and impacts directly the time required for understanding and locating the source of the error. As well, this standardization may lead to an automatic treatment of validation messages by IT tools.

It would be desirable to reach certain level of harmonization in the presentation of validation results with agreed validation metrics. More about metrics on validation could be found in the second part of this handbook: Metrics for a data validation procedure. The quality measures could be used as standard reporting devices which are routinely calculated.

The purpose of this phase is gathering the statistics on validation outcomes to assess the quality of data sets and quality of validation rules.

Data, programs and the corresponding metadata have to be documented and archived if the process should be repeated or if new methods will be tested for data sets. It is desirable to have common approach for validation procedure to keep validation rules in one place maintained and supported continuously, friendly users’ application and specification written in understandable language for different users of the application.

Activity descriptions

* Data are checked against the validation rules. Validate data against predefined validation rules.
* Summarising results. It depends on the user of the results (staff, management or methodologist).

**Review phase**

This phase is aimed at continuous improvement of validation process efficacy and data quality. During the review phase needs for new design elements are established. This phase includes identification of problems using feedback from the users and other stakeholders and analysing outcomes from the execution phase. The identified problems are prioritised and dealt with in the design phase.

Examples of revisions are:

Improvement of validation rules due to:

- Replacing those that detect few errors by others more powerful

- Replacing those that ‘mislead’: detect errors that are not real errors

- Increase efficiency of validation rules

- Improvements in validation rules: detecting more possible errors

- Changes in the data file or regulations

Changes in the validation process originated by:

- Changes in validation tools

- Changes in file formats

- Improving efficiency

Changes in the validation workflow due to:

- Better assignment of responsibilities in validation tasks

- Efficiency gains in the chain

Activity descriptions

* Analysis of feedback from stakeholders. Feedback gathered in previous phases.
* Analysing of outcomes from the execution phase. Identified potential problems, errors, discrepancies, detected systematic problems are analysed in order to decide whether validation rules should be reviewed.
* Identifying and prioritising problems.

**Metrics for data validation**

**Introduction to the metrics for data validation**

The objective of the good design of a validation procedure (including possible rules) is to achieve a satisfactory quality that would permit the statisticians to have a reasonable confidence that the outcome of the validation process is free of important errors. Moreover, the cost of that process should not be disproportionate subject to satisfying some requirements of efficiency (fast, low detection of false errors) and completeness (most true errors are detected).

Designing and maintaining a validation procedure is a dynamic learning process as it can be improved through the experiences drawn from the checking of successive data sets. For example, evaluation of the existing validation rules could be periodically performed on the basis of the validation results.

To facilitate such evaluations, it is helpful to be able to quantify the current performance of the procedure in some way. Examples include counting violations per rule, or per record, the number of rules used in a procedure (as they tend to proliferate in practice), number of redundant rules and so on. Any such quantification is a metric (quality indicator) of the data validation procedure.

Metrics should measure the efficacy and efficiency of a data validation procedure. The indicators may refer either to a single rule or to the whole data validation procedure.

The metrics that will be discussed in the coming subsections may be distinguished in:

- Indicators taking into account only validation rules (properties of validation rules)

- Indicators taking into account only observed data

- Indicators taking into account both observed and reference data (e.g., imputed data, simulated data).

The first two are generally used to fine tune the data validation procedure, for instance in the design phase and by using pilot survey.

The metrics of the third type are used to obtain a more precise measure of the effectiveness of a data validation procedure, but are dependent on the method chosen to obtain amended plausible data, or synthetic data. The method is composed of an error localization procedure and an imputation procedure.

Evaluation of validation rules can be done by looking at their efficacy, i.e., the capacity of reaching the target objective. However, when evaluating a validation rule, it should be considered also its capacity to find important errors. These two aspects, already defined with the term ‘severity’, are to be considered jointly when evaluating the efficacy of a validation rule. As an example, let’s look at a balance edit x+y=z and at a ratio edit a<x/y<b defined to check outliers. The first one

addresses an error with a probability 1, the second, by the definition of outliers, identifies potential errors (probability less than 1) but given that the values are in the tails, it can be useful to find important errors. A balanced analysis of these two aspects is necessary to evaluate the efficacy of validation rules.

**Metrics for data validation rules and data structure**

Since data validation is an important and intrinsic part of statistical production and data exchange, it is worthwhile to regard the building blocks of data validation, the data validation rules, as object of study. In particular, it is interesting for reasons related to quality of both the statistical process and the data to have insight into the (over)completeness, effectiveness, redundancy and feasibility and complexity of a set of rules. The ability to describe such properties, regardless of to what data set the rules are applied can help practitioners to weed out redundancies and understand the implications of combinations of rules.

There seems to be little literature in official statistics that aims directly at these goals, although a number of the basic techniques that are necessary for such investigations have been discussed in the context of other problems where rule manipulation is a basic requirement. In particular, several techniques have been developed in the context of error localization (see De Waal et al., 2011 and references therein). Since the literature on these techniques is limited to what are commonly called ‘in-record’ (cross-variable) validation rules, the current section is limited to these rules as well3 except when otherwise indicated. For example, a rule such as

Profit + Cost = Total Revenue,

is included in the discussion, but a rule stating

The average price of this month must not differ from last month’s price by more than 50%,

is not covered in this discussion. That is to say, the latter example is covered to the extent in which they can be represented as in-record rules. One may always create a data set where the average price of the current and previous month are stored in a single record. However, all the information used to compute the averages is then lost and not part of computing the outcome of the validation rule. For example, in case the rule is violated there is no way based on that rule and average data alone to localize the record or records that cause the violation (if any).

Since the explicit literature on rule maintenance and properties is limited, so is the discussion in the current section of this handbook. However, we have attempted to gather a number of approaches and techniques that appear useful when describing rule sets – but necessarily on a shallow level. In the following Sections we cover techniques to determine the completeness, redundancy, consistency, and complexity of a set of rules.

10.1 Completeness

With completeness, we mean the extent to which prior knowledge about a data set has been expressed in terms of a set of validation rules. Since the term ‘prior knowledge’ is hard to quantify, it will in general be difficult to find tools or methods to systematically asses completeness. Worse than that, with the current state of practice it is hard to encode all domain knowledge in (hard) validation rules. Knowledge that is easily encoded includes restrictions that follow from physical or logical facts such as: `age cannot be negative’, or: `the profit must be smaller than or equal to revenue’. Knowledge that is hard to encode in rules include facts or events whose influence on the values recorded in a data set can only be stated in imprecisely. For example, although it is clear that a takeover, fusion, or acquisition influences the cost structure of a company, it is difficult to quantify this in anyway precisely. Another example, consider political choices and economic circumstances that influence the purchasing power of certain households. An expert judging household records or aggregates may take knowledge of such situations into account, but quantifying is exceedingly difficult. With these limitations in mind, in this section we focus primarily on knowledge that can be stated with precision in terms of in-record validation rules.

Two obvious completeness-related problems may occur. The first is incompleteness, by which we mean that there are restrictions, implied by physics or logic that have not been stated explicitly or implicitly by a set of rules. The second is over-completeness, which means that the set of rules is too restrictive and value combinations that are actually valid are unjustly excluded.

**10.1.1 Peer review**

The basis of finding out whether a set of rules in- or overcompletes with respect to the knowledge of a domain is by knowledgeable peers. Given a rule set that is produced by an expert or team of experts, one can follow two approaches for judging completeness.

In the first approach a second team of peers (or peer) sets up a set of rules independently of the original team. The differences between the rule sets: rules that occur in one, but not in the other can then be discussed in terms of necessity and validity. It is important that the focus of such a discussion is on real world assumptions that underlie the rule sets rather than the way they have been stated. In fact, we will show in section 10.2 that it is possible to determine whether two sets of rules are equivalent (in a sense to be stated more precisely), so such discussions can be avoided.

In the second approach, a rule set and its documentation is reviewed by a (team of) peer(s). For each rule the underlying assumption is judged against expert knowledge, and it is checked whether these assumptions are made sufficiently clear by either the form of the rule or clarity of documentation. One might score each rule, indicating to what extent the underlying assumption is hard, amenable to change over time, or is `soft’ for example because it depends on a threshold of which the value is to some extent subjective. Such indicators can be used to judge as to how often a (subset of) the rule set must/should be revised.

**10.1.2 Formal methods**

On the more formal side, one may investigate to what extent the variables in a data set are covered by the rules in a set. Recall that we may associate with an in-record validation rule a validation function that takes a record of data and returns a value in {0,1}, where 0 indicates failure (the rule is violated) and 1 indicates success (the rule is satisfied). For example, given a record with the variables profit (𝑝), staff cost (𝑠), total cost (𝑐), and total revenue (𝑡). We can define the following rules 𝑝+𝑐=𝑡 𝑠≤𝑐

For the rule +𝑐=𝑡 , we can define the validation function 𝑣 as 𝑣:𝑅4→{0,1}

𝑣(𝑝,𝑠,𝑐,𝑡)=𝑖𝑓𝑝+𝑐=𝑡𝑡ℎ𝑒𝑛1𝑒𝑙𝑠𝑒0.

Now, we say that a variable occurs in a validation function (or rule) when the value of the validation function can change when the variable is changed. In this example, the variable 𝑠 does not occur in 𝑣 and variables 𝑝,𝑐,∧,𝑡 do occur in 𝑣.

As a first metric for coverage, one check for each variable whether it occurs in at least one of the explicitly defined rules. It is reasonable to assume that for each measured or observed variable occurs in at least one check. In the example, all four variables occur in at least one rule.

It is tempting to extend this check and to tabulate the number of occurrences for each variable. However, one must be careful when interpreting such tables since these numbers are not in general uniquely defined. For instance from the rule set above the numbers of occurrences is

(𝑝=1,𝑠=2,𝑐=3,𝑡=2). However, we can solve 𝑐 from the demand that 𝑝+𝑐=𝑡 and substitute it in the demands 𝑠≤𝑐 and 𝑐≥0. This yields the equivalent4 rule set 𝑝+𝑐=𝑡 𝑠≤𝑡−𝑝

𝑠≥0 (1b) 𝑡≥0 𝑡−𝑝≥0.

Counting the occurrences now yields (𝑝=3,𝑠=2,𝑐=1,𝑡=4). In fact, the number of rules can be variable as well. Consider the following subset of rule set (1) 𝑠≤𝑐 𝑠≥0 𝑐≥0

The rule 𝑐≥0 is redundant, since there is no way that rule 𝑠≤𝑐 and 𝑠≥0 can be satisfied for negative 𝑐. In other words, rule set (1) is also equivalent to the set 𝑝+𝑐=𝑡 𝑠≤𝑐

The above indicates that counting variable occurrences (or rules) has little meaning, unless one somehow declares a standard (possibly minimal or irredundant) way for formulating rules. This conclusion generalizes to any data validation rule set since it really only depends on the question whether logical deductions can be made from a set of rules. As far as the author is aware, there is currently no method or algorithm described in literature that allows one to derive a unique representation from any set of in-record validation rules.

**10.2 Redundancy**

A set of validation rules divides the space of all possible records into a valid, or acceptance region and an inacceptable region. Suppose we have a set of validation rules 𝑉. We say that a rule is redundant in 𝑉 if removing it from the set of rules does not alter the acceptance region.

There are two reasons to remove redundancies from a set of rules. First, redundancy removal yields a more compact and elegant rule set that expresses the necessary restrictions in some sense minimally. Secondly, the time and/or memory consumption of some algorithms that make use of rule sets can depend strongly on the number of rules provided. One example is the branch-and-bound error localization algorithm of De Waal and Quere (2003), whose time and memory consumption grow exponentially with the number of rules if redundancies are not taken care of at runtime.

The downside of removing redundant rules is that they may be less understandable by experts. For example, as stated above, the rules 𝑠≤𝑐 𝑠≥0,

together imply that 𝑐≥0. This becomes however only apparent after some reasoning. In practice, rule sets with more than 100 (in)equalities are not uncommon so one can hardly expect a domain expert to produce or interpret an irredundant rule set. Moreover, when interpreting the output of a validation step (that is, the set of Boolean values that result from confronting data with a set of restrictions) one would like to connect that output directly with the defined rules, rather than with a reduced set.

This means there is a trade-off between the mathematical rigor of obtaining an absolutely irreducible set of rules and a user-friendly set that is more directly connected to the assumptions formulated by domain experts. A reasonable compromise is to let domain experts formulate the rules in a formal language that is close to their own, but to remove redundancies automatically as much as possible when this is beneficial for the algorithms making use of such rules.

**10.2.1 Methods for redundancy removal**

Several methods for removal of redundant constraints have been described in literature. A recent comparative overview in the context of linear programming is given by Paulraj and Sumathi (2010). As an illustration on how to approach redundancy removal, we follow Daalmans (2015) and describe shortly a method that has been described by Chmeiss et al. (2008) and also by Felfernig (2014).The idea of the method depends on the ability to find inconsistencies (see next Section).

Consider a set 𝑉={𝑣1,𝑣2,…,𝑣𝑚} of validation rules. Suppose that one of the 𝑣𝑖∈𝑉 is redundant, that is, it must always yield 1 when the other rules yield 1 (we also say that 𝑣𝑖 is implied by the other rules). Then, if we replace 𝑣𝑖 in 𝑉 with the opposite rule, the resulting set of rules becomes infeasible. That is, the acceptance region is the empty set: no record can ever satisfy all rules in this adapted set. As an example, consider again the following set of rules, which we now give names, for convenience. 𝑉={𝑣1:=𝑠≤𝑐,𝑣2:=𝑠≥0,𝑣3:=𝑐≥0}.

Now,𝑣2 and 𝑣3 imply that any combination (𝑠,𝑐) must lie in the first quadrant of the plane, while 𝑣1implies that only points on and above the line 𝑐=𝑠 are valid. Let us now replace 𝑣3 with its opposite and define 𝑉={𝑣1:=𝑠≤𝑐,𝑣2:=𝑠≥0,¬𝑣3:=𝑐<0}.

Now, 𝑣2 and ¬𝑣3 imply that every point must lie in the fourth quadrant of the 𝑠−𝑐 plane, but not on the 𝑠 axis. This conflicts with 𝑣1 which implies that every point (𝑠,𝑐) must be on or above the line 𝑐=𝑠.

This idea, that replacing an implied rule with its negation yields an infeasible rule set is fully general, and is thus not limited to numerical examples as given above. Given that we are able to establish the feasibility of a set of rules, the following procedure for redundancy detection readily presents itself (the backslash indicates set difference, and ¬ indicates negation).

**10.3 Feasibility**

A rule set is called feasible, or informally also consistent, when the acceptance region defined by a rule set is nonempty. Infeasibility occurs for instance when a rule set contains a rule that is contradictory in itself, for example the rule that states 𝑥>𝑥 is clearly contradictory. As a second example, consider the following rule set has rules that are perfectly feasible by themselves but their combination is contradictory: {𝑥>1,𝑥<0}

Clearly, there is no number 𝑥 that can satisfy both rules. In practice, rule sets can be much more complicated than this and contradictions rarely present themselves in such a clear form.

Now, consider a general set of validation rules 𝑉={𝑣1,𝑣2,…,𝑣𝑚}. We may think of these rules as functions that take a data record, say 𝑟, and return a value𝑣𝑖(𝑟) in {0,1}, where 0 means the rule is violated and 1 means the rule is satisfied. We emphasize that 𝑟 is not necessarily numeric, but comes from a domain 𝐷 containing all possible records that may have been measured. For example, consider a survey where we ask 𝑛 persons about their age and whether they are employed. The domain for a single record might be described by 𝐷=𝑅×{𝑦𝑒𝑠,𝑛𝑜}. An example record is 𝑟=(38,𝑦𝑒𝑠) and we have the rule set

**10.3.1 Methods for finding inconsistencies**

There are two common strategies for determining the feasibility of a set of in-record feasibility rules. Both strategies have in common that the rules must be expressible in a single general form. To denote this general form, we first write a general record so that all the categorical variables 𝑐𝑖 come first, and the numerical variables 𝑥𝑖 come next.

Now, we assume that rules can be written in the form (De Waal, 2002) 𝑖𝑓𝑐∈𝐹𝑡ℎ𝑒𝑛𝑥∈{𝑥∈𝑅𝑛:Σ𝑎𝑖𝑛𝑖=1𝑥𝑖≤0}.

Here, 𝐹 is a subset of combinations of categories and the 𝑎𝑖 are real coefficients. It can be shown [see e.g. De Waal (2002), De Waal et al (2011)] that many commonly occurring in-record validation rules can be written in such a form. Amongst others, this form includes multivariate conditions on categorical variables, linear equality and/or inequality restrictions and conditional rules involving both categorical and numerical linear restrictions. A slight generalization was recently formulated by De Jonge and Van der Loo (2014) in the context of software for error localization problems.

The first strategy to determine feasibility is based on methods for simplifying rule sets by eliminating variables. For example, consider the rule set {𝑥≥1,𝑥<0}.

We write this set as a system of linear inequalities, and add the two demands together as follows

−𝑥<1

𝑥<0+

The resulting demand contains one variable less than the two original rules and it is an obvious contradiction. Since the original system implies a contradiction, the system must be infeasible.

This example can be generalized in the following ways. First of all, variables can always be eliminated from systems of inequalities through a procedure called Fourier-Motzkin elimination (see e.g. Williams 1986). Also, it can be shown that if a set of linear inequalities is infeasible, repeated Fourier-Motzkin elimination will always lead to a simple contradictions such as 0<1. For rules concerning categorical variables, a procedure called multivalent resolution (Hooker, 2000) can be used to eliminate a variable, which again can be repeated to generate obvious contradictions if the original set was infeasible to begin with. Finally, De Waal (2002) shows how these elimination methods may be combined to include rules of the general form of Equation (2).

The second strategy relies on mixed-integer programming (MIP), and existing mixed-integer problem solving software. The idea is to make the set of validation rules part of the restrictions in a specific optimization problem which is then fed to a MIP solver. If no solution can be found by the solver the set of rules is contradictory. See Daalmans (2015) and references therein for examples.

Advantages of the first strategy include reliability and numerical stability. Fourier-Motzkin elimination is available in free software, such as the R-package editrules (De Jonge and Van der Loo, 2011) or in a more basic interface in the lintools package (Van der Loo and De Jonge, 2015). However, variable elimination methods can be both computationally intensive and may consume

**10.4.1 Information needed to evaluate a rule**

The first notion of complexity is related to the variety of information that is necessary to evaluate a validation rule. In the simplest case, a rule can be evaluated by comparing a single data value with a fixed range of values. Stepping up a level in complexity, we find rules that compare a data value by one or more other values, for example in balance edits, or rules where an observation at time 𝑡 is compared to an observation at an earlier time 𝑡−1. Going up in complexity we find rules where a value is compared with (functions of) ranges of other values. For example, a value can be compared with a location estimator of a different variable, computed from the current data set.

This notion of the ‘variety of information’ necessary to compute a validation rule is precisely the principle behind the formal typology that has been described in the first part of this handbook, in Section 5. There, a minimal set of four labels are identified that characterize a data point: the Universe (or domain, type of statistical objects), the time of measurement, the actual unit on which a measurement was performed and the variable that was measured. The levels of complexity then correspond to the number of labels that must be varied in order to compute the validation function.

We refer to Section 5 for further explanation and elaborated examples.

**11 Metrics for a data validation procedure**

Once a validation process has been designed for a given dataset, it needs to be tested in order to assess its suitability for the observed data and to find the best set of techniques and/or parameters able to optimize its performance. The testing aims at evaluating the performance of the validation rules in terms of efficacy (ability to achieve the objectives) and efficiency. Based on the results from testing, some of the design decisions and/or parameters could be revisited to optimize quality. Two types of analysis can be performed: in a first evaluation on the impact of the process, based only observed data are used, in a second type of evaluation both observed and reference data (true, or synthetic) are used.

11.1 Indicators on validation rule sets derived from observed data

Investigation based only on observed data may provide useful insight for tuning the validation rules. Indicators taking into account observed data and treated data, i.e. edited data, are not included in this section. In fact, our aim is to evaluate the rule introduced in a data validation procedure, and according to the definition, the editing step is not a part of the process. Indicators based on the comparison of observed and treated data are useful to assess the efficacy of an editing and imputation procedure. This topic will be addressed in section 11.2. See also EDIMBUS (2007) and references therein.

The indicators based only on the observed data exploit only information related to the application of rules to data. They can be calculated overall or per variable/observation, but as stated in the Chapter 9, they refer only to in-record rules. The extension to inter-record rules is an interesting topic that deserves further studies.

In the following list, some examples for validating numerical and logical variables are reported:

1. Number of failed records

2. Minimum number of variables to be changed in order to make records pass the given set of rules (cardinality of solution)

3. Counts of records that passed, missed and failed for each rule

4. Distribution of records that passed missed and failed k rules

5. Counts of rules applications of status pass, miss, fail

6. Counts of records of status pass, miss or fail for which field j contributed to the overall record status

7. Ratio of missing records against failed record counts (NAs/number of failed records) – a measure of (non-)responsiveness against erroneous records.

8. Difference (or percent) of 1-6 indicators between current and previous period observed datasets.

In the following three examples in the context are reported.

Example 1

Quality indicators (metrics) on data validation that could be computed separately, e.g. for each statistical survey derived from observed data based on questionnaires:

* The number and share (%) of statistical questionnaires validated due to respondent or data entry mistakes compared to the total number of questionnaires (a questionnaire is considered as erroneous if at least one fatal edit rule has been unsatisfied).
* The number and share (%) of statistical questionnaires validated due to respondent mistakes compared to the total number of questionnaires.
* The number and share (%) of statistical questionnaires validated by the specialists compared to the total number of questionnaires.
* The number and share (%) of values validated by the specialists divisions compared to the total number of entered values.
* The number and share (%) of flagged objects by rule 𝑟 compared to the total number of objects

A discussion on the use in practice of those indicators should be included. For instance, a high number of failures of a validation rule may suggest either the presence of a systematic error or the inappropriateness of the validation rule.

**Example 3**

One more example more aiming on evaluation of data validation process that are mostly based on processing data from editing.

Table 6. Example of indicators that are also done in combination with data editing (provided by Statistics Sweden)

|  |  |  |  |
| --- | --- | --- | --- |
| Statistics Sweden) **Metric** | | **Description** | |
| **O1** Share of objects with error signals | 𝑁𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝑜𝑏𝑗𝑒𝑐𝑡𝑠 𝑤𝑖𝑡ℎ 𝑒𝑟𝑟𝑜𝑟 𝑠𝑖𝑔𝑛𝑎𝑙𝑁𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝑜𝑏𝑗𝑒𝑐𝑡𝑠 𝑡ℎ𝑎𝑡 ℎ𝑎𝑠 𝑟𝑢𝑛 𝑡ℎ𝑟𝑜𝑢𝑔ℎ 𝑡ℎ𝑒 𝑣𝑎𝑙𝑖𝑑𝑎𝑡𝑖𝑜𝑛 𝑐ℎ𝑒𝑐𝑘𝑠 | | Illustrate the scope of validation, should be calculated for several survey instances. |
| **O2** Share of edited objects | 𝑁𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝑒𝑑𝑖𝑡𝑒𝑑 𝑜𝑏𝑗𝑒𝑐𝑡𝑠 𝑁𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝑜𝑏𝑗𝑒𝑐𝑡𝑠 𝑡ℎ𝑎𝑡 ℎ𝑎𝑠 𝑟𝑢𝑛 𝑡ℎ𝑟𝑜𝑢𝑔ℎ 𝑡ℎ𝑒 𝑣𝑎𝑙𝑖𝑑𝑎𝑡𝑖𝑜𝑛 𝑐ℎ𝑒𝑐𝑘𝑠 | | Illustrate the effects of validation. |
| **O6 Total hit rate** | 𝑁𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝑜𝑏𝑗𝑒𝑐𝑡𝑠 𝑤𝑖𝑡ℎ 𝑒𝑟𝑟𝑜𝑟 𝑠𝑖𝑔𝑛𝑎𝑙𝑠 𝑎𝑛𝑑 𝑒𝑑𝑖𝑡𝑠 𝑁𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝑜𝑏𝑗𝑒𝑐𝑡 𝑤𝑖𝑡ℎ 𝑒𝑟𝑟𝑜𝑟 𝑠𝑖𝑔𝑛𝑎𝑙𝑠 | | Illustrate the efficiency of validation at large. |
| **K1** Share of objects with error signals per validation check | 𝑁𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝑒𝑟𝑟𝑜𝑟 𝑠𝑖𝑔𝑛𝑎𝑙𝑠 𝑓𝑟𝑜𝑚 𝑣𝑎𝑙𝑖𝑑𝑎𝑡𝑖𝑜𝑛 𝑐ℎ𝑒𝑐𝑘 𝑘𝑁𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝑜𝑏𝑗𝑒𝑐𝑡𝑠 𝑡ℎ𝑎𝑡 ℎ𝑎𝑠 𝑟𝑢𝑛 𝑡ℎ𝑟𝑜𝑢𝑔ℎ 𝑡ℎ𝑒 𝑣𝑎𝑙𝑖𝑑𝑎𝑡𝑖𝑜𝑛 𝑐ℎ𝑒𝑐𝑘𝑠 | | Illustrate the exposure of each validation check. |
| **K2** Hit rate per validation check | 𝑁𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝑜𝑏𝑗𝑒𝑐𝑡𝑠 𝑤𝑖𝑡ℎ 𝑒𝑟𝑟𝑜𝑟 𝑠𝑖𝑔𝑛𝑎𝑙𝑠 𝑓𝑟𝑜𝑚 𝑣𝑎𝑙𝑖𝑑𝑎𝑡𝑖𝑜𝑛 𝑐ℎ𝑒𝑐𝑘 𝑘 𝑡ℎ𝑎𝑡 ℎ𝑎𝑣𝑒 𝑏𝑒𝑒𝑛 𝑒𝑑𝑖𝑡𝑒𝑑𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝑜𝑏𝑗𝑒𝑐𝑡𝑠 𝑤𝑖𝑡ℎ 𝑒𝑟𝑟𝑜𝑟 𝑠𝑖𝑔𝑛𝑎𝑙𝑠 𝑓𝑟𝑜𝑚 𝑣𝑎𝑙𝑖𝑑𝑎𝑡𝑖𝑜𝑛 𝑐ℎ𝑒𝑐𝑘 𝑘 | | Illustrate the efficiency of each validation check. |
| **V7** Shaer of edited objects per variable | 𝑁𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝑜𝑏𝑗𝑒𝑐𝑡𝑠 𝑤ℎ𝑒𝑟𝑒 𝑣𝑎𝑟𝑖𝑎𝑏𝑙𝑒 𝑋 ℎ𝑎𝑠 𝑏𝑒𝑒𝑛 𝑒𝑑𝑖𝑡𝑒𝑑 𝑁𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝑜𝑏𝑗𝑒𝑐𝑡𝑠 𝑤𝑖𝑡ℎ 𝑣𝑎𝑙𝑢𝑒𝑠 𝑓𝑜𝑟 𝑣𝑎𝑟𝑖𝑎𝑏𝑙𝑒 𝑋 | | Illustrate problematic variables and possible measurement errors. |
| **V8 Net effect of validation on variable estimates** | 𝐸𝑠𝑡𝑖𝑚𝑎𝑡𝑒 𝑏𝑎𝑠𝑒𝑑 𝑜𝑛 𝑢𝑛𝑒𝑑𝑖𝑡𝑒𝑑 𝑣𝑎𝑙𝑢𝑒𝑠𝐸𝑠𝑡𝑖𝑚𝑎𝑡𝑒 𝑏𝑎𝑠𝑒𝑑 𝑜𝑛 𝑒𝑑𝑖𝑡𝑒𝑑 𝑣𝑎𝑙𝑢𝑒𝑠-1 | | Illustrate the effect of validation on the produced statistics. |

For a validation rule set of m positivity validation rules and n user-specified validation rules with no redundant validation rules, a total of m + n + 1 status codes is assigned to each record. Redundant validation rules should be removed from the validation rules group by this stage, but if they are still included, the number of status codes to be generated will be reduced because redundant validation rules do not appear in the tables. The following procedure is performed independently for each data record.

 One status code is assigned to the data record for each validation rule, including the positivity rules. There are m + n of these codes in total. The status is:

o PASS, if the record passes the validation rule,

o MISS, if the record has one or more missing fields involved in the validation rule, or

o FAIL, if the record fails the validation rule because of one or more non-missing values.

- If the overall record status is PASS then the record passed all validation rules, all fields must be good and all fields are assigned a PASS status for the purposes of this table.

- If the overall record status is MISS, then MISS and PASS are the only possible values for validation rule status. The fields involved in the validation rule set with status MISS are assigned a MISS and the fields not involved in any validation rule set with status MISS are assigned a NOT APPLICABLE status.

- If the overall record status is FAIL, then at least one validation rule status must be FAIL and one or more validation rule status may be MISS. The variables involved in validation rule set with FAIL validation rule status are assigned a FAIL and the variables not involved in any validation rule with a FAIL validation rule status are assigned a NOT APPLICABLE status.

11.2.1 True values as reference data

A data validation procedure can be viewed, at least at record level, as a classification process, according to which records are classified as erroneous or not. Thus, we could adopt some usual measures for classification procedures like for instance confusion matrix or ROC curve. However, while the validation procedure often refers to complex data objects (typically records), errors are generally referred to single items. Thus, evaluating efficacy of a validation procedure should also involve the analysis of changes of single values resulting from the procedure. In order words, the metrics should be extended to the classification of single variables as erroneous or not. Unfortunately, this makes the evaluation task somewhat ambiguous. In fact, apart the special case of rules involving only one variable (domain rules), generally application of a validation rule does not imply any localisation step, that is, any decision about which variable(s) is responsible for the edit failure. This is typically the objective of a separate localisation procedure, often based on some mathematical algorithm (e.g. Fellegi-Holt methodology). Thus, assessing the capability of correctly classifying single values as erroneous or not would actually result in evaluating the whole data editing procedure including both data validation and error localisation. Furthermore, the quality of the final data to be used for statistical purposes also depends on the method(s) used to replace values flagged as erroneous with plausible ones (imputation). Thus, efficacy indicators related to the accuracy of the final data would also include evaluation of the imputation methodology.

From the last observations, it follows that it is difficult to assess the efficacy and efficiency of a data validation procedure independently from the evaluation of other phases of the editing and imputation process. Nevertheless, it is always possible to evaluate a validation procedure in terms of its capability of correctly identifying records containing at least one error (“erroneous records”). Of course, this approach does not distinguish records containing a different number of errors.

Using symbols C and E to denote absence or presence of error respectively, we can define in the usual manner the “confusion table” as the 2X2 cross table of the true versus predicted values for the error indicator:

Number of errors at item level and severity

The indicators so far introduced are appropriate for evaluating classification procedures with binary outcomes, where the objects to be classified are records of a dataset, and the predicted dichotomous variable is “presence of at least an error” in each record. As already mentioned, this metrics does not take into account differences in the number of erroneous values within the records. Moreover, in case of quantitative variables, it could be desirable to have metrics capable of distinguishing large and small discrepancies between true and observed values (i.e., measures of “severity” of the errors). As noticed above the number of incorrect values in a single record, and in case of numerical variables the error magnitude, can only be estimated if some localization procedure is used. Localizing erroneous variables however, is not part of the validation phase, whose only outcome is the split of the data in validated and not validated records. A possible approach to overcome these difficulties is to assume (ideally) that “perfect” procedures for error localization and imputation are available, so that once a record is flagged as “not valid”, the true value can be restored with certainty. In this manner, we could evaluate the validation procedure by comparing the observed data with the true data. Classical indicators for the evaluation of editing procedures can be used. Below, some examples are provided.

Indicators for both categorical and numerical variables

Single variable (Y):

Indicators based on the confusion matrix (PPV, ACC, TNR,..), where the categories for the binary variables are “actual presence of error in variable Y” (rows) and “flag associated with the output of the validation procedure” (columns). This confusion matrix corresponds to considering the variable Y incorrect in each not validated record. For instance, if Y is the variable AGE, the counts in the corresponding confusion matrix refer to number of times that records are validated or not and variable AGE is correctly reported or not.

Overall:

The same indicators as in the previous case but now applied to all the data items simultaneously. The confusion matrix corresponds to considering all variables of not validated records as erroneous.

Severity indicators for numerical variables

In case of numerical variables it would be desirable to have some indicator measuring the importance of the errors that cause records to be rejected by the validation procedure. When true data are available, natural way to define “importance” is in terms of absolute difference between observed and true value. Again, in order to have meaningful metrics, we should consider all the variables in not validated records as erroneous and derive indicators by comparing the dataset

Appendix A: List of validation rules

Table A.2.3.2: Examples of checking rules for logical validation and consistency

|  |  |
| --- | --- |
| **Check** | **Comment and remarks regarding the level and type of the check (c.f. 2.3.1)** |
| **Simple Identity Check** | |
| *In a sheep survey :* “Milk production” must be equal to “milk disposal”  *Year*  Year = Year (in the filename)  *Population:* The total for each population (persons, households, families, dwellings) is checked to be consistent throughout the process.  *Enterprises:* An enterprise included in the admin data must be part of the predetermined population (from the Business Register)  *Number of livestock and animal production survey:* Number of animal at the end of reference period == number of animal at the beginning of following reference period. | *Mirror check.*  This might be an example for levels 4 or 5, if "production" and "disposal" data come from different data sources or even from different data providers  This might be an example for level 2 or higher, if data on persons, households, families and dwellings are stored in different files and come perhaps form different data sources  This might be an example for level 2 or higher, if “admin data” and “business register” are organized as different data sets, or even managed by different units in the institute.  *Mirror check.*  Level 2, if we assume that data from different periods are stored in different files |
| **Simple range check - bounds fixed** | |
| *Working hours (monthly)*: “Hours worked” must be between 0 and 168  *Number of inhabitants for country LU*:  Total number of inhabitants should be in the range 100,000 – 1,000,000. | |