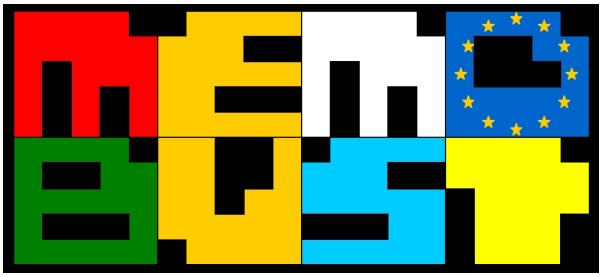

This module is part of the

Memobust Handbook

on Methodology of Modern Business Statistics

26 March 2014

Method: Deductive Editing

Contents

General section

1. Summary

Data collected for compiling statistics frequently contain obvious systematic errors; in other words, errors that are made by multiple respondents in the same, identifiable way (see "Statistical Data Editing – Main Module"). Such a systematic error can often be detected automatically in a simple manner, in particular in comparison to the complex algorithms that are needed for the automatic localisation of random errors (see the method module "Statistical Data Editing – Automatic Editing"). Furthermore, after a systematic error has been detected, it should be immediately clear which adjustment is necessary to resolve it. For we know, or think we know with sufficient reliability, how the error came about.

A separate deductive method is needed for each type of systematic error. The exact form of the deductive method varies per type of error; there is no standard formula. The difficulty with using this method lies mainly in determining *which* systematic errors will be present in the data, before these data are actually collected. This can be studied based on similar data from the past. Sometimes, such an investigation can bring systematic errors to light that have arisen due to a shortcoming in the questionnaire design or a bug in the processing procedure. In that case, the questionnaire and/or the procedure should be adapted. To limit the occurrence of discontinuities in a published time series, it can be desirable to 'save up' changes in the questionnaire until a planned redesign of the statistic, and to treat the systematic error with a deductive editing method until that time.

2. General description of the method

2.1 Introduction to deductive editing

In this module, we focus on methods for detecting and treating so-called systematic errors. As mentioned in "Statistical Data Editing – Main Module", a systematic error is commonly defined as an error with a structural cause that occurs frequently between responding units. A well-known type of systematic error is the so-called *unit of measurement error* which is the error of, for example, reporting financial amounts in units instead of the requested thousands of units.

Systematic errors can introduce substantial bias in aggregates, but once detected, systematic errors can easily be treated because the underlying error mechanism is known. It is precisely this knowledge of the underlying cause that makes the treatment of systematic errors different from random errors. Treating systematic errors based on knowledge of the underlying error mechanism is called *deductive editing*. Systematic errors can often be identified by examining frequently occurring edit rule failures. Deductive methods are therefore mainly effective for data for which many edit rules have been defined.

Deductive editing of systematic errors is an important first step in the editing process. It can be done automatically and reliably at virtually no costs. Moreover, the rest of the editing process can proceed more efficiently after the systematic errors have been resolved. Deductive editing is in fact a very effective and probably often underused editing approach.

Any systematic error for which the cause is understood with sufficient certainty can be resolved deductively. In the case of incorrect assumptions about the error mechanism, however, deductive

editing may introduce a bias in the estimators. In practice, a deductive method might also be used to resolve certain random errors, for reasons of efficiency, provided that the introduced bias is negligible. An example of this is the deductive resolution of rounding errors (see Scholtus, 2011).

De Waal and Scholtus (2011) make a further distinction between *generic* and *subject-related* systematic errors. Errors of the former type occur for a wide variety of variables in a wide variety of surveys and registers, where the underlying cause is always essentially the same. Apart from the unit of measurement error, other examples include *simple typing errors*, such as interchanged or mistyped digits (Scholtus, 2009) and *sign errors*, such as forgotten minus signs or interchanged pairs of revenues and costs (Scholtus, 2011). For an example that involves a simple typing error, see Section 3.2 below. Generic errors can often be detected and treated automatically by using mathematical techniques.

Subject-related systematic errors are specific to a particular questionnaire or survey. They may be caused by a frequent misunderstanding or misinterpretation of some question such as reporting gross values rather than net values. Another example is that, for some branches of industry, staff is frequently classified as belonging to an incorrect department of the responding enterprise. Subject-related systematic errors are usually detected and treated by applying correction rules that have been specified by subject-matter experts.

The remainder of this text is organised as follows. Section 2.2 further discusses the use of correction rules for subject-related systematic errors. Section 2.3 discusses techniques that treat possibly the most notorious of generic systematic errors, the unit of measurement error. Section 2.4 discusses methods for identifying new systematic errors.

2.2 Correction rules for subject-matter related errors

Subject-matter related errors can often be detected and treated by means of deterministic checking rules. Such rules state which variables are to be considered erroneous when the edits are failed in a certain way. Often, deterministic checking rules also describe how the erroneous variables should be adjusted. In that case, these rules are commonly referred to as *correction rules*.

The general form of a correction rule is as follows:

if (*condition*) **then** (*correction*).

Here, *condition* indicates a combination of values in a record that is not allowed. Subsequently *correction* describes the adjustment that is made to the record to resolve the inconsistency.

An example of a correction rule is:

if (*Number of Temporary Employees* > 0 **and** *Costs of Temporary Employees* = 0)
then *Number of Temporary Employees* := 0. (1)

This rule detects an inconsistency that occurs when a business reports to have employed temporary staff without reporting associated costs. In this example, the inconsistency is treated deductively by making the number of temporary employees equal to zero.

In general, a correction rule is intended to resolve an inconsistency that can be resolved in a unique way on logical and/or content-related grounds, under a certain assumption. If the assumption is valid, the deductive editing method always reproduces the true values. For instance, the correction rule (1)

operates under the assumption that the variable *Costs of Temporary Employees* is reported more accurately than the variable *Number of Temporary Employees*. Making such assumptions in a valid way generally requires subject-matter knowledge and knowledge of the data collection process.

Correction rules are attractive because of their simplicity. However, they may only be used when no important nuances are lost with such a simple approach. If the data do not satisfy the assumptions made, then deductive editing may lead to biased estimators. For instance: if in the above example it happens that some businesses actually forget to report the costs of temporary employees, then, after applying the correction rule (1), we may underestimate the total number of temporary employees for businesses in the target population.

Another potential drawback of using correction rules is that a large collection of correction rules may be difficult to maintain, especially when the collection has grown over a long period of time. In particular, it then becomes difficult to grasp the effects of adding a new correction rule, or removing an old one, or changing the order in which the rules are applied to the data. For this reason, it is usually not recommended to try to treat all possible errors in a rule-based manner, because this would require a very complex set of correction rules. Broadly speaking, deductive editing should be limited to the treatment of systematic errors only. For the treatment of random errors, there exist other methods that are more powerful and less difficult to maintain (see "Statistical Data Editing – Automatic Editing").

2.3 The unit of measurement error

Business surveys usually contain instructions to the reporter that all financial amounts must be rounded to thousands of euros (dollars, pounds, etc.), that all quantities must be rounded to thousands of units, et cetera. Some respondents ignore these instructions and, consequently, report values that are a factor 1000 larger than they actually mean. It is clear that, if these *thousand-errors* are not corrected, the resulting estimates for the figures to be published will be too high. The thousand-error is a commonly encountered special case of the more general unit of measurement error, which occurs whenever respondents report values that are consistently too high or too low by a certain factor.

We refer to a *uniform* unit of measurement error if all variables (of a certain type) in a record are too large by the same factor. It is known that, in practice, records with *partial* unit of measurement errors also occur. A partial unit of measurement error could arise, for instance, if several departments of a business each fill in part of a questionnaire independently. Partial unit of measurement errors are generally more difficult to detect than uniform ones.

Traditional methods for detecting unit of measurement errors usually work by comparing one or more reported amounts with reference values. The type of reference data used and the way in which the comparison takes place varies per statistic and per statistical office. Examples of reference data are: a statement from the same respondent from an earlier period, the median value of a number of similar respondents in an earlier period or the same period, and available register data about the respondent.

A widely used method computes the ratio of the unedited value and the reference value. If this ratio is larger than a lower bound, or lies between certain bounds, then it is concluded that the unedited value contains a unit of measurement error. Once a unit of measurement error has been detected, it is treated deductively by dividing all relevant amounts by an appropriate factor. It is often assumed for convenience that all unit of measurement errors are uniform.

For instance: in the Dutch Short Term Statistics, thousand-errors are detected as follows (Hoogland et al., 2011). The total turnover indicated by the respondent for period *t*, say x_t, is compared to the turnover from the most recent period for which a statement from the respondent is available, up to a maximum of six previous periods. The stated turnover for this earlier period must also not be equal to zero. A thousand-error is detected in x_t if the following applies:

$$|x_t| > 300 \times |x_{t-i}|, \qquad \text{for some } i \in \{1,\ldots,6\}.$$

If no data from the respondent from an earlier period are available, then the median of the turnover from the previous period in the stratum of the respondent is used instead. The stratification is based on economic activity and number of employees. A thousand-error is detected in x_t if the following applies:

$$|x_t| > 100 \times \text{stratum median}(x_{t-1}).$$

If a thousand-error is detected by either formula, then it is resolved by dividing the total turnover and all the sub-items by 1000.

Table 1 shows an example of a record with a thousand-error that was found in this way.

Table 1. Example of a uniform thousand-error

	reference data	unedited data	data after treatment
first sub-item turnover	3,331	3,148,249	3,148
second sub-item turnover	709	936,142	936
total turnover	4,040	4,084,391	4,084

It should be noted that the above-described method assumes that the reference value is not affected by unit of measurement errors. Thus, the reference value should either be based on previously edited data, or it should be calculated in a way that is robust to the presence of (some) unit of measurement errors.

Clearly, the choice of bounds in the detection method for unit of measurement errors is important. There is a trade-off here between the number of missed errors (observations that are supposedly correct, but actually contain unit of measurement errors) and the number of false hits (observations that supposedly contain unit of measurement errors, but are actually correct). If previously edited data are available, then a simulation study can be conducted to experiment with different bounds. See Pannekoek and De Waal (2005) for an example of such a simulation study.

In manual editing, unit of measurement errors are often detected using a graphical aid. As an illustration, Figure 1 shows a scatter plot of unedited values of turnover (on the *y* axis) against reference values (on the *x* axis), with both variables plotted on a logarithmic scale (using the logarithm to base 10). A cluster of thousand-errors can clearly be identified near the line $y = x + 3$.

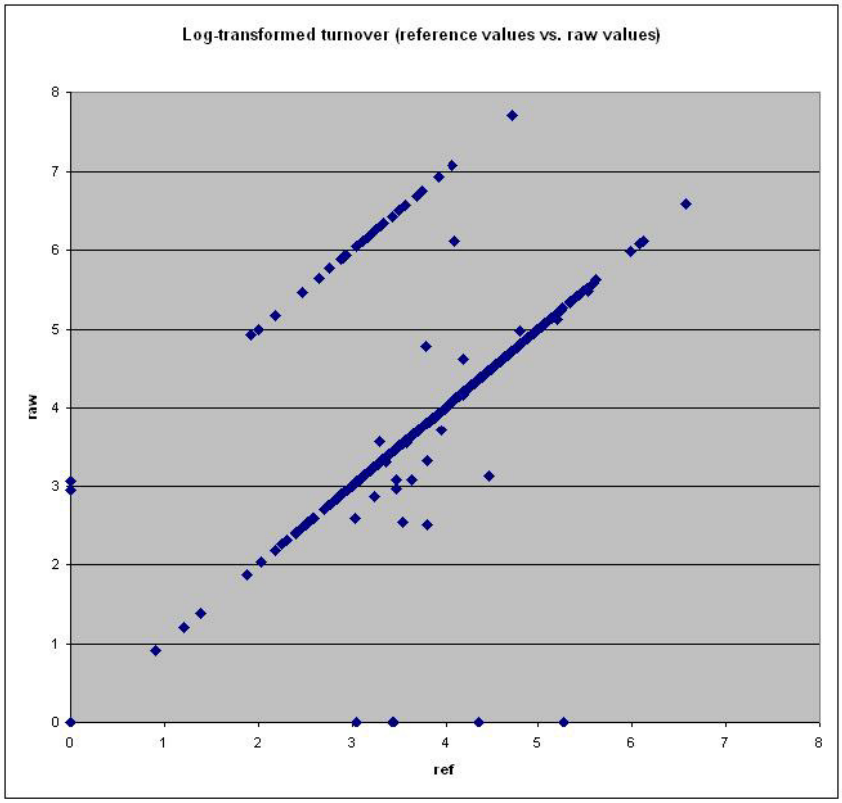

Figure 1. A scatter plot displaying thousand-errors on a logarithmic scale

Elaborating on this graphical approach, Al-Hamad et al. (2008) proposed an alternative automatic method for detecting unit of measurement errors. They considered the difference between the number of digits in the unedited value and the reference value:

$$diff = \left| \lceil \log_{10} x \rceil - \lceil \log_{10} x_{ref} \rceil \right|, \tag{2}$$

where $\lceil a \rceil$ denotes the smallest integer larger than or equal to a. Using (2), different types of unit of measurement errors may be detected by identifying records with a certain value of *diff*. For example, a thousand-error corresponds to *diff* = 3. It should be noted that this method can also detect unit of measurement errors in the reference data, because the absolute value is taken in (2).

Di Zio et al. (2005) proposed a more complex method for detecting unit of measurement errors, by explicitly modeling both the true data and the error mechanism. They used a so-called finite mixture model to identify different clusters within the data set. Each cluster contains records that are affected by a particular type of unit of measurement error; there is also one cluster of records without unit of measurement errors.

Compared with the traditional methods for detecting unit of measurement errors, the approach of Di Zio et al. (2005) has several interesting features. First, it does not require reference data, because the model is fitted directly to the unedited data. However, reference values may also be included in the model if they are available. Second, the method provides diagnostic measures of its own performance, which can be used to identify observations with a significant probability of being misclassified. A selection of doubtful cases may then be checked by subject-matter experts. Finally, this method provides a natural way to detect partial unit of measurement errors. A drawback of the method is that it

may not always be possible to fit an appropriate model to the data set, especially for data sets with many variables or irregular structures. Di Zio et al. (2007) consider an extension of this approach that can accommodate more general models.

2.4 Identifying new systematic errors

New systematic errors can be identified by analysing edit rule failures. If an edit rule is frequently failed, this can be an indication of the presence of a systematic error in the relevant variables. A further analysis of the records that fail the edit rule, in which the questionnaire is also examined, can bring the cause of the error to light. Once the error has been identified, it is generally quite simple to draw up a deductive method to automatically detect and treat the error.

Detecting new systematic errors can only take place once sufficient data have been collected. The results are therefore usually too late to be used in the production process of the current survey cycle. If the analysis produces new deductive editing methods, then these can be built into the editing process for the data in the next survey cycle.

As far as systematic errors are concerned, prevention is better than cure. Sometimes it is possible to improve the design of the questionnaire so that far fewer respondents make a certain type of error. If many respondents make the same kind of error, this can in fact be an indication that a certain question is not presented clearly enough. In some cases, it is also possible to adapt the processing procedure to ensure that a certain processing error no longer arises. In principle, this approach should be preferred to that of making deductive adjustments afterwards. However, because there are practical objections to the constant adaptation of the questionnaire, one may choose initially to build in a deductive editing method, and to use the accumulated knowledge of systematic errors later in a redesign of the questionnaire. (See also the module "Repeated Surveys – Repeated Surveys".) Moreover, some systematic errors appear to be impossible to prevent, no matter how well the questionnaire is designed. This is, for instance, the case with the unit of measurement error.

To illustrate the identification of a new systematic error, we consider the data collected in 2001 for the Dutch Structural Business Statistics for Wholesale. In this data set, there are (among many other variables) five variables on labour costs, which should satisfy the following edit rule:

$$x_1 + x_2 + x_3 + x_4 = x_5 . \tag{3}$$

Here, x_5 represents the variable *total labour costs*. The other four variables are the sub-items of this total. Table 2 shows several records that do not satisfy edit rule (3).

Table 2. Examples of inconsistent partial records in the Dutch SBS for Wholesale 2001

	record 1	record 2	record 3	record 4
x_1	1,100	364	1,135	901
x_2	88	46	196	134
x_3	40	34	68	0
x_4	42	0	42	0
x_5	170	80	306	134

It is striking that, for all records in Table 2, it holds that $x_2 + x_3 + x_4 = x_5$. This suggests that these reporters have ignored the first sub-item x_1 in the calculation of x_5. A closer look at the questionnaire (see Figure 2) reveals why this could have happened: there is a gap between the answer box for x_1 and the other boxes. As a result, from the design of the questionnaire alone, it is ambiguous whether x_1 should be part of the sum or separate from the rest. Most respondents understand from the context what the intention is, but in several dozen records, we found the same error as in Table 2.

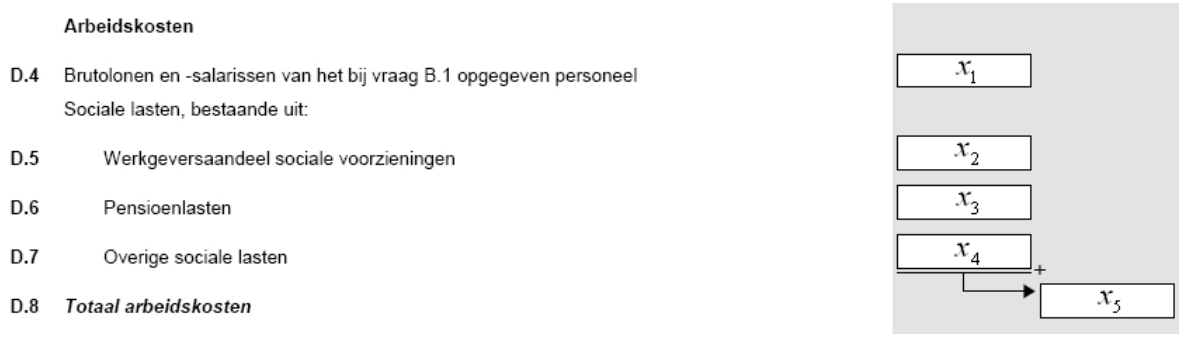

Figure 2. Part of the questionnaire used for the Dutch SBS Wholesale (until 2005)

We can draw up a deductive method that resolves this error. A more structural solution consists of removing the cause of the error by adapting the questionnaire. This has already been done: the questionnaire from Figure 2 was replaced for the Dutch Structural Business Statistics of 2006. On the new questionnaire, the answer boxes are spaced evenly.

3. Preparatory phase

4. Examples – not tool specific

4.1 Example: Correction rules for the statistic Building Objects in Preparation[1]

The Dutch quarterly statistic Building Objects in Preparation (BOP) follows the development of the total construction value of new contracts at architectural firms in the Netherlands. In 2007, a new editing process was designed for this statistic.

When filling in the BOP questionnaire, the reporter must answer several questions about each building object separately. The reporter must tick a box indicating whether the building object concerns a residence (*r*), a combined-purpose building (*c*; this means that the building is used for other purposes as well as residential purposes) or neither of these (*o* for other). Another question concerns *n*, the total number of dwellings in the building. For a combined-purpose building, the percentage of floor area intended for residential use (*p*) is also requested.

[1] This example is adapted from a report written in Dutch by Mark van der Loo and Jeroen Pannekoek (Statistics Netherlands).

The statement contains an error if zero, two, or three of the boxes for *r*, *c*, and *o* have been ticked. In that case, the type of building object has not been clearly specified. In certain situations, this error can be treated deductively based on the values of *n* and *p*.

If the value indicated for *n* is greater than zero and if, moreover, *p* is equal to 100% or is not filled in, then it is obvious that the building object is a residence. If *n* is larger than zero and furthermore if *p* is not equal to 0 or 100%, it is obvious that the building object is a combined-purpose building. And, finally, if neither *n* nor *p* has been filled in, or if they have been given the value of 0, then it is highly probable that the building object falls in the category 'other'. These interpretations follow from the assumption that the statement must be rendered correct by changing as few values as possible.

We write $r = \mathsf{T}$ if the box for residence has been ticked, and otherwise $r = \mathsf{F}$, and we do the same for *c* and *o*. The following correction rule expresses the deductive assertions made in the previous paragraph in formal notation:

if $(r,c,o) \in \{ (\mathsf{T},\mathsf{T},\mathsf{T}) , (\mathsf{T},\mathsf{T},\mathsf{F}) , (\mathsf{T},\mathsf{F},\mathsf{T}) , (\mathsf{F},\mathsf{T},\mathsf{T}) , (\mathsf{F},\mathsf{F},\mathsf{F}) \}$
then
if (p = 'empty' **or** $p = 100\%$) **and** $n > 0$
then $(r,c,o) = (\mathsf{T},\mathsf{F},\mathsf{F})$
if $0\% < p < 100\%$ **and** $n > 0$
then $(r,c,o) = (\mathsf{F},\mathsf{T},\mathsf{F})$
if (p = 'empty' **or** $p = 0\%$) **and** (n = 'empty' **or** $n = 0$)
then $(r,c,o) = (\mathsf{F},\mathsf{F},\mathsf{T})$.

This is a small part of the editing process for the statistic BOP.

In the implementation of the editing process for BOP, the derivation of the correction always takes place separately from the actual application of the correction. Initially, in the above example, only an indicator is created that specifies for each record whether a deductive correction is applicable, and if so, which one. Only in the next step are the values of *r*, *c* and *o* changed in the record. As such, the editing process is transparent, so that it is clearly visible afterwards exactly what changes have been made to each record.

4.2 Example: Simple typing errors

We consider a fictitious survey in which the values of *Turnover*, *Costs*, and *Profit* are asked from businesses. By definition, these variables are related through the following edit rule:

$$Turnover - Costs = Profit. \tag{4}$$

The first column of Table 3 shows a record that is inconsistent with respect to (4). The inconsistency can be resolved by adapting any one of the three variables. Moreover, under the assumption that only one variable contains an error, its true value can be computed by inserting the observed values of the other variables into equation (4). The other columns of Table 3 show the three consistent versions of the original record that can be produced by adapting one of the variables (the adapted value is shown in bold in each column).

Table 3. Example of a record with a simple typing error

	record	adjustment 1	adjustment 2	adjustment 3
Turnover	252	**315**	252	252
Costs	192	192	**129**	192
Profit	123	123	123	**60**

Intuitively, the solution in which *Costs* is adapted is the most attractive, since it has the nice interpretation that two adjacent digits were interchanged by mistake. That is to say, it seems much more probable that the true value of 129 was changed to 192 at some point during the collection and processing of the data, than the case that 315 was changed to 252 or 60 to 123. Therefore, we could draw up the following rule for deductive editing: if a record does not satisfy (4), but it can be made consistent by interchanging two adjacent digits in one of the observed values (and, moreover, this can be done in a unique way), then the inconsistency should be treated in this way.

Interchanging two adjacent digits is an example of a simple typing error. Other examples include:

- adding a digit (for example, writing '1629' instead of '129');
- omitting a digit (for example, writing '19' instead of '129');
- replacing a digit (for example, writing '149' instead of '129').

Common features of all simple typing errors are that they only affect one value at a time, and that they produce an observed erroneous value which is related to the unobserved true value in a way that is easy to recognise.

In the example from Table 3, the simple typing error could be detected by using the fact that the variables should satisfy edit rule (4). In general, a survey may contain variables that are related by many equalities and also by other types of edit rules. Moreover, the equalities may be interrelated, so that variables have to satisfy different edit rules simultaneously. Scholtus (2009) described a deductive method for detecting and treating simple typing errors in this more general setting.

Simple typing errors are generic errors, because they occur in many different surveys and they are not content-related. This type of error is easy to make and can therefore occur frequently in practice. A review of data from the Dutch Structural Business Statistics for Wholesale in 2007 revealed, for example, that nearly 10% of all inconsistencies in linear equalities could be explained by one of the four typing errors mentioned above (Scholtus, 2009).

5. Examples – tool specific

The R package `deducorrect`, which can be downloaded for free at http://cran.r-project.org, contains an implementation of deductive editing methods for several generic errors:

- sign errors and interchanged values;
- simple typing errors (as defined in Section 3.2);
- rounding errors (very small inconsistencies with respect to equality constraints).

The underlying methodology is described by Scholtus (2011) for sign errors and rounding errors, and by Scholtus (2009) for simple typing errors. To illustrate the use of `deducorrect`, we work out an example. Consider a data set of 11 variables that should satisfy the following edit rules:

$$\begin{cases} x_1 + x_2 = x_3 \\ x_2 = x_4 \\ x_5 + x_6 + x_7 = x_8 \\ x_3 + x_8 = x_9 \\ x_9 - x_{10} = x_{11} \end{cases}$$

The following record is inconsistent with respect to these edit rules; in fact, it does not satisfy the second, fourth, and fifth constraints:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
1452	116	1568	161	323	76	12	411	19979	1842	137

We shall use the `deducorrect` package to treat this record for simple typing errors. First, we load the package:

```
> library(deducorrect)
```

Next, we create an object of type “editmatrix” containing the system of edit rules:

```
> E <- editmatrix( c("x1 + x2 == x3",
+                    "x2 == x4",
+                    "x5 + x6 + x7 == x8",
+                    "x3 + x8 == x9",
+                    "x9 - x10 == x11") )
```

We also have to read in the record that we want to treat as a data frame:

```
> x <- data.frame( x1 = 1452, x2 = 116, x3 = 1568, x4 = 161,
+                  x5 = 323, x6 = 76, x7 = 12, x8 = 411,
+                  x9 = 19979, x10 = 1842, x11 = 137 )
```

To check whether simple typing errors can be found in this record, we use the function `correctTypos` provided by the package:

```
> sol <- correctTypos(E, x)
```

The object `sol` is a list which contains the results of the search for simple typing errors. We first check the status of the record:

```
> sol$status
     status
1 corrected
```

The status ‘corrected’ means that the record could be made consistent with respect to all edit rules by only treating simple typing errors. Other possible statuses are: ‘valid’ for a record that was consistent in the first place, ‘invalid’ for an inconsistent record in which no typing error could be detected, and ‘partial’ for a record that could be made consistent with respect to some, but not all edit rules by treating simple typing errors.

The list `sol` also contains the adjusted version of the record and a table of the suggested adjustments:

```
> sol$corrected
    x1  x2   x3  x4  x5 x6 x7  x8   x9  x10 x11
1 1452 116 1568 116 323 76 12 411 1979 1842 137

> sol$corrections
  row variable   old  new
1   1       x4   161  116
2   1       x9 19979 1979
```

Thus, `correctTypos` has detected two simple typing errors in this example: the value of x_4 should be 116 instead of 161 (interchanged adjacent digits), and the value of x_9 should be 1979 instead of 19979 (added digit). By treating these errors, a consistent record is obtained with respect to all edit rules.

We refer to Van der Loo et al. (2011) for more details on the `deducorrect` package.

6. Glossary

For definitions of terms used in this module, please refer to the separate "Glossary" provided as part of the handbook.

7. References

Al-Hamad, A., Lewis, D., and Silva, P. L. N. (2008), Assessing the Performance of the Thousand Pounds Automatic Editing Procedure at the ONS and the Need for an Alternative Approach. Working Paper, UN/ECE Work Session on Statistical Data Editing, Vienna.

De Jong, A. (2002), Uni-Edit: Standardized Processing of Structural Business Statistics in the Netherlands. Working Paper, UN/ECE Work Session on Statistical Data Editing, Helsinki.

De Waal, T. and Scholtus, S. (2011), Methods for Automatic Statistical Data Editing. Paper presented at the 2011 KSS International Conference on Statistics and Probability, Busan.

Di Zio, M., Guarnera, U., and Luzi, O. (2005), Editing Systematic Unity Measure Errors through Mixture Modelling. *Survey Methodology* **31**, 53–63.

Di Zio, M., Guarnera, U., and Rocci, R. (2007), A Mixture of Mixture Models for a Classification Problem: The Unity Measure Error. *Computational Statistics & Data Analysis* **51**, 2573–2585.

Hoogland, J., van der Loo, M., Pannekoek, J., and Scholtus, S. (2011), *Data Editing: Detection and Correction of Errors*. Methods Series Theme, Statistics Netherlands, The Hague.

Pannekoek, J. and de Waal, T. (2005), Automatic Edit and Imputation for Business Surveys: the Dutch Contribution to the EUREDIT Project. *Journal of Official Statistics* **21**, 257–286.

Scholtus, S. (2009), Automatic Correction of Simple Typing Errors in Numerical Data with Balance Edits. Discussion Paper 09046, Statistics Netherlands, The Hague.

Scholtus, S. (2011), Algorithms for Correcting Sign Errors and Rounding Errors in Business Survey Data. *Journal of Official Statistics* **27**, 467–490.

Van der Loo, M., de Jonge, E., and Scholtus, S. (2011), Correction of Rounding, Typing, and Sign Errors with the deducorrect Package. Discussion Paper 201119, Statistics Netherlands, The Hague.

Specific section

8. Purpose of the method

Detecting and treating errors in a deductive manner

9. Recommended use of the method

1. The method should be used, in principle, only for detecting and treating systematic errors.
2. Deductive editing is most effective when it is applied at the very beginning of the editing process, before any other form of editing has been used.

10. Possible disadvantages of the method

1. Deductive editing should only be used to treat errors for which the error mechanism is known with sufficient reliability. Deductive adjustments based on invalid assumptions can produce biased estimators.
2. It may be difficult to maintain a large collection of deterministic correction rules over a long period of time. In particular, it becomes difficult to grasp the consequences of adding or removing a correction rule, or changing the order in which the rules are applied, when faced with a large collection of rules.

11. Variants of the method

1. Each type of systematic error requires its own particular variant.

12. Input data

1. A data set containing unedited microdata.
2. If relevant, a data set containing reference data

13. Logical preconditions

1. Missing values
    1. Allowed, but an assumption has to be made on their interpretation (e.g., “consider all missing values to be equal to zero unless evidence to the contrary is found”).
2. Erroneous values
    1. Allowed; in fact, the object of this method is to detect and treat some of them.
3. Other quality related preconditions
    1. n/a
4. Other types of preconditions
    1. n/a

14. Tuning parameters

1. If relevant, a collection of edit rules for the microdata.

2. Other parameters, depending on the particular variant / type of error.

15. Recommended use of the individual variants of the method

1. n/a

16. Output data

1. A data set containing partially edited microdata, which is an updated version of the first input data set.

17. Properties of the output data

1. Ideally, the data set should contain no more systematic errors, only random errors.

18. Unit of input data suitable for the method

Incremental processing by record

19. User interaction - not tool specific

1. User interaction is not needed during an execution of deductive editing.

20. Logging indicators

1. All adjustments that are introduced by each deductive editing method should be flagged as such. This helps to keep the editing process transparent and it also provides input for future analyses of the editing process itself.

21. Quality indicators of the output data

1. The quality of deductive editing can be assessed in a simulation study. This requires a data set that has been edited by experts to a point where the edited data may be considered error-free. In the simulation study, the original data are edited again using deductive editing methods. The quality of a deductive editing method may then be measured in terms of its success in detecting systematic errors in the original data set.
2. Alternatively, one could also perform a simulation study by introducing artificial systematic errors into an existing data file. The quality of a deductive editing method may then be measured in terms of its success in identifying these artificial errors.

22. Actual use of the method

1. Several forms of deductive editing are used in the production process for Structural Business Statistics at Statistics Netherlands (see De Jong, 2002).

Interconnections with other modules

23. Themes that refer explicitly to this module

1. Repeated Surveys – Repeated Surveys
2. Statistical Data Editing – Main Module

24. Related methods described in other modules

1. Statistical Data Editing – Automatic Editing

25. Mathematical techniques used by the method described in this module

1. n/a

26. GSBPM phases where the method described in this module is used

1. GSBPM Sub-process 5.3: Review, validate and edit

27. Tools that implement the method described in this module

1. R package `deducorrect`

28. Process step performed by the method

Statistical data editing

Administrative section

29. Module code

Statistical Data Editing-M-Deductive Editing

30. Version history

Version	Date	Description of changes	Author	Institute
0.1	22-12-2011	first version	Sander Scholtus	CBS (Netherlands)
0.2	20-04-2012	improvements based on Swedish review	Sander Scholtus	CBS (Netherlands)
0.2.1	16-07-2013	adjusted to new template; minor improvements	Sander Scholtus	CBS (Netherlands)
0.3	04-09-2013	minor improvements	Sander Scholtus	CBS (Netherlands)
0.3.1	09-09-2013	preliminary release		
1.0	26-03-2014	final version within the Memobust project		

31. Template version and print date

Template version used	1.0 p 4 d.d. 22-11-2012
Print date	21-3-2014 18:11