Automatic correction of simple typing errors in numerical data with balance edits

09

Sander Scholtus

Discussion paper (09046)

Statistics Netherlands

The Hague/Heerlen, 2009

Explanation of symbols

.	= data not available
*	= provisional figure
x	= publication prohibited (confidential figure)
–	= nil or less than half of unit concerned
–	= (between two figures) inclusive
0 (0,0)	= less than half of unit concerned
blank	= not applicable
2007–2008	= 2007 to 2008 inclusive
2007/2008	= average of 2007 up to and including 2008
2007/'08	= crop year, financial year, school year etc. beginning in 2007 and ending in 2008
2005/'06–2007/'08	= crop year, financial year, etc. 2005/'06 to 2007/'08 inclusive

Due to rounding, some totals may not correspond with the sum of the separate figures.

Publisher
Statistics Netherlands
Henri Faasdreef 312
2492 JP The Hague

Prepress
Statistics Netherlands - Grafimedia

Cover
TelDesign, Rotterdam

Information
Telephone +31 88 570 70 70
Telefax +31 70 337 59 94
Via contact form: www.cbs.nl/information

Where to order
E-mail: verkoop@cbs.nl
Telefax +31 45 570 62 68

Internet
www.cbs.nl

ISSN: 1572-0314

6008309046 X-10

Automatic correction of simple typing errors in numerical data with balance edits

Sander Scholtus

Summary: Data collected for the production of structural business statistics consist of a large number of numerical variables, with many mathematical relations between them. These relations are specified in the form of consistency checks, called edit rules or edits. Edits are used to detect errors that occur in the data set. One of the purposes of the editing process is to remove these errors. For the sake of efficiency, it is desirable to automate the detection and correction of errors as much as possible, and this need has led to the development of general methods for automatic error localisation. In this paper, we present a simple device for the correction of inconsistencies in balance edits (linear equalities) due to simple typing errors, such as interchanged digits. The occurrence of these particular errors can be predicted accurately from the unedited data. Since general methods for automatic error localisation do not use this information, it seems advantageous to correct simple typing errors automatically in a separate step.

Keywords: automatic editing, deductive correction, typing errors, structural business statistics

1 Introduction

In the theory of editing, a distinction is often made between systematic errors and random errors. Systematic errors are made consistently by different respondents, because of a structural cause. The best-known example is the unity measure error, where the respondent reports amounts that are too high by a constant factor, usually a power of ten. Other examples of systematic errors are presented in Scholtus (2008). Deductive algorithms can often be used to detect and correct these errors automatically. Random errors are caused by non-structural problems during data collection, such as simple typing errors.

Because of these errors, an unedited record may violate one or several consistency checks, known as edit rules or edits. Two examples of edits are:

$$turnover - costs = profit \tag{1}$$

and

$$number\ of\ employees \geq 0. \tag{2}$$

A common approach to resolve random errors is to search for a minimal set of variables that can be changed such that all edits become satisfied simultaneously. This is known as the Fellegi-Holt paradigm; cf. Fellegi and Holt (1976). The paradigm is often used in a generalised form, where each variable is given a reliability weight and the objective becomes to minimise the sum of reliability weights of the variables to be changed. Algorithms for data editing based on the generalised Fellegi-Holt paradigm have been implemented in various software packages, such as Statistics Canada's GEIS and Banff, the U.S. Bureau of the Census's SPEER and DISCRETE, and Statistics Netherlands' SLICE.

A typical editing process may contain two automatic error localisation steps. In the first step, errors are treated that are known to occur frequently and that can be resolved by a tailor-made deductive algorithm. For instance, unity measure errors can be treated in this step. In the second step, all remaining errors are removed by solving a mathematical optimisation problem based on the Fellegi-Holt paradigm. Thus, it is tacitly assumed that all systematic errors have been removed in the first step. In particular, the editing process for structural business statistics at Statistics Netherlands contains these two steps; cf. De Jong (2002). Of the two, the second localisation step is by far the most computationally intensive.

An edit in the form of a linear equality, such as (1), is called a balance edit. Van de Pol et al. (1997) observe that if a random error produces a violated balance edit, the unedited data may contain information on the nature of the error that is not used by the Fellegi-Holt paradigm. In particular, this may happen if the error only causes a small perturbation in a true value, for instance when two digits are interchanged. For example, suppose that a record contains the following values:

turnover	*costs*	*profit*
353	283	115

and thus violates edit (1). Assuming that this is the only edit, and that all variables have the same reliability weight, an application of the Fellegi-Holt paradigm yields

three equivalent solutions, namely that one of the variables should be changed to obtain consistency. That is, either *turnover* is changed to $283+115=398$, or *costs* is changed to $353-115=238$, or *profit* is changed to $353-283=70$. From these values, it appears that changing *costs* to 238 is probably the correct solution, since it has the nice interpretation that two digits in the true value were interchanged by accident. The other solutions do not have a clear interpretation. However, the Fellegi-Holt paradigm does not use this information.

For the case that the data should satisfy exactly one balance edit, Van de Pol et al. (1997) describe a method to incorporate this kind of information in the reliability weights. In the example above, the reliability weight of the variable *costs* would be lowered by a certain factor. Applying the generalised Fellegi-Holt paradigm then yields the desired solution. The practical use of this method is limited, because in real-world applications with numerical data, the Fellegi-Holt paradigm is usually applied with a much larger set of edits, including more than one balance edit. Hence, to our best knowledge this method has not been applied in practice.

The purpose of the present paper is to extend the method of Van de Pol et al. (1997) to the more general situation where records have to satisfy a system of inter-related edits. Moreover, we describe how automatic corrections can be generated, which makes it possible to apply the method in the first localisation step mentioned above rather than the second. This decreases the amount of computational work needed in the second step. The remainder of this paper is organised as follows: Section 2 introduces the types of errors we hope to treat with our method; Section 3 describes the method; an example is discussed in Section 4; some refinements are suggested in Section 5; a practical application of the method is discussed in Section 6; finally, Section 7 concludes the paper.

2 Simple typing errors

We shall consider the following simple typing errors in this paper: (a) interchanging two adjacent digits; (b) adding a digit; (c) omitting a digit; (d) replacing a digit. In this section, we give a formal description of these four errors. A common feature of these types of errors is that they always affect one variable at a time. This is not true of errors in general; consider for instance the unity measure error. Another common feature of these types of errors is that they result in an observed erroneous value, which is related to the unobserved correct value in an easily recognisable way. Again, the same cannot be said of general errors.

To formally describe the errors introduced above, we first have to choose a numerical system to represent survey values. We take a pragmatic view and assume that the decimal system is used, although this restriction is not necessary. We also assume throughout this paper that all variables are integer-valued. Thus, every observed value can be written in the form

$$x = \pm \sum_{j=0}^{M} \xi_j \cdot 10^j, \tag{3}$$

where ξ_j denotes the j-th digit of x. Note that digits are numbered from right to left in the standard notation and that the rightmost digit is called the 0-th digit. In (3), M is a

positive integer such that $|x| < 10^{M+1}$ for all observed values. Although a theoretical interpretation is lacking, it is not difficult to find a suitable value of M in practice, because observed values are stored in computer memory using a limited number of bits.

A simple typing error can be seen as a function $f : \mathbb{Z} \to \mathbb{Z}$ acting on the true value x. Because of a random error, the value $f(x)$ is observed instead of x. We can write down explicit expressions for the functions that describe the four types of errors mentioned above.

(a) Interchanging two adjacent digits: this is described by the family of functions $f_{\mathrm{ic}}(x;k)$, with

$$f_{\mathrm{ic}}(x;k) = x + \xi_k \cdot (10^{k+1} - 10^k) + \xi_{k+1} \cdot (10^k - 10^{k+1}), \tag{4}$$

for $k = 0, \ldots, M-1$. The function $f_{\mathrm{ic}}(x;k)$ interchanges the digits ξ_k and ξ_{k+1}. For instance: $f_{\mathrm{ic}}(4627;1) = 4267$.

(b) Adding a digit: this is described by the family of functions $f_{\mathrm{a}}(x;k,\xi)$, with

$$f_{\mathrm{a}}(x;k,\xi) = \sum_{j=0}^{k-1} \xi_j \cdot 10^j + \xi \cdot 10^k + \sum_{j=k+1}^{M} \xi_{j-1} \cdot 10^j, \tag{5}$$

for $k = 0, \ldots, M$ and $\xi = 0, \ldots, 9$. The function $f_{\mathrm{a}}(x;k,\xi)$ adds a digit ξ at the k-th position. For instance: $f_{\mathrm{a}}(4627;1,8) = 46287$. Applying this function to x only makes sense if $\xi_M = 0$.

(c) Omitting a digit: this is described by the family of functions $f_{\mathrm{o}}(x;k)$, with

$$f_{\mathrm{o}}(x;k) = \sum_{j=0}^{k-1} \xi_j \cdot 10^j + \sum_{j=k+1}^{M} \xi_j \cdot 10^{j-1}, \tag{6}$$

for $k = 0, \ldots, M$. The function $f_{\mathrm{o}}(x;k)$ omits the digit ξ_k from x. For instance: $f_{\mathrm{o}}(4627;1) = 467$.

(d) Replacing a digit: this is described by the family of functions $f_{\mathrm{r}}(x;k,\xi)$, with

$$f_{\mathrm{r}}(x;k,\xi) = x + (\xi - \xi_k) \cdot 10^k, \tag{7}$$

for $k = 0, \ldots, M$ and $\xi = 0, \ldots, 9$. The function $f_{\mathrm{r}}(x;k,\xi)$ replaces the k-th digit of x by ξ. For instance: $f_{\mathrm{r}}(4627;1,8) = 4687$.

We remark that the parameter k has the interpretation that $f(x;k)$ shares its first digits with x up to and including ξ_{k-1}, but possibly has a different digit in the k-th position. Also, to interpret the outcome of $f_{\mathrm{ic}}(x;k)$ as a typing error, it has to hold that $x \geq 10^{k+1}$. To interpret the outcome of $f_{\mathrm{a}}(x;k,\xi)$ as a typing error, it has to hold that $x \geq 10^{k-1}$. Finally, to interpret the outcome of $f_{\mathrm{o}}(x;k)$ or $f_{\mathrm{r}}(x;k,\xi)$ as a typing error, it has to hold that $x \geq 10^k$.

We assume that for each respondent there exists an unobserved true record $\mathbf{y}$, which satisfies all edits. Several error mechanisms act on the values in this record, producing an observed record $\mathbf{x}$, which is available in digital form at the statistical institute, but possibly contains errors. We assume that these error mechanisms operate independently of each other, and that each variable is affected by at most one error mechanism.

3 Theory for automatic correction of simple typing errors

3.1 Analysing violated and satisfied edits

For now, we assume that the variables $\mathbf{x} = [x_1, \ldots, x_n]'$ have to satisfy only balance edits $e_1, \ldots, e_m$. The r-th balance edit e_r states that

$$a_{r,1}x_1 + \cdots + a_{r,n}x_n = 0, \tag{8}$$

where all coefficients $a_{r,i}$ are integers. Together, these edits can be written as $\mathbf{Ax} = \mathbf{0}$, where $\mathbf{A}$ is an $m \times n$-matrix of coefficients and $\mathbf{0}$ is the m-vector of zeros. We discuss an extension of the method that also handles other types of edits in Section 5.

Each edit defines a three-way partition of $\{1, \ldots, n\}$:

$$I_1^{(r)} = \{i : a_{r,i} > 0\}, \qquad I_2^{(r)} = \{i : a_{r,i} < 0\}, \qquad I_3^{(r)} = \{i : a_{r,i} = 0\}, \tag{9}$$

for $r = 1, \ldots, m$. Edit e_r can be written as

$$\sum_{i \in I_1^{(r)}} a_{r,i}x_i = -\sum_{i \in I_2^{(r)}} a_{r,i}x_i. \tag{10}$$

When $i \in I_3^{(r)}$, we say that x_i is not involved in edit e_r. The complement $\bar{I}_3^{(r)} = I_1^{(r)} \cup I_2^{(r)}$ contains the indices of all variables involved in edit e_r.

Similarly, each variable defines a partition of $\{1, \ldots, m\}$:

$$R_1^{(i)} = \{r : a_{r,i} > 0\}, \qquad R_2^{(i)} = \{r : a_{r,i} < 0\}, \qquad R_3^{(i)} = \{r : a_{r,i} = 0\}, \tag{11}$$

for $i = 1, \ldots, n$. The complement $\bar{R}_3^{(i)} = R_1^{(i)} \cup R_2^{(i)}$ contains the indices of all edits that involve x_i. We assume throughout that each variable is involved in at least one edit, i.e. $\bar{R}_3^{(i)} \neq \emptyset$ for all i, since a variable that is not involved in any edits can be ignored during editing.

Given an observed record $\mathbf{x}$, it is possible to compute, for each edit, two partial sums:

$$s_1^{(r)} = \sum_{i \in I_1^{(r)}} a_{r,i}x_i, \qquad s_2^{(r)} = -\sum_{i \in I_2^{(r)}} a_{r,i}x_i, \qquad r = 1, \ldots, m. \tag{12}$$

The record violates edit e_r, and we write $\phi(r) = 1$, if $s_1^{(r)} \neq s_2^{(r)}$ (see (10)). Otherwise, the record satisfies edit e_r and we write $\phi(r) = 0$. Thus, the set of edits is split into two groups:

$$E_1 = \{r : \phi(r) = 1\}, \qquad E_2 = \{r : \phi(r) = 0\}. \tag{13}$$

The edits with indices in E_1 are violated by the current record, whereas the edits with indices in E_2 are satisfied.

Finally, we define the following subset of the variables:

$$I_0 = \bigcap_{r \in E_2} I_3^{(r)} = \left\{ i : E_2 \subseteq R_3^{(i)} \right\}. \tag{14}$$

This subset has the following interpretation: it is the index set of variables that are not involved in any edit that is satisfied by the current record. In other words, all edits

that involve a variable from I_0 are violated by the current record. When searching for simple typing errors, we only want to perform corrections that increase the number of satisfied edits, without causing previously satisfied edits to become violated. This provision implies that the only variables we can safely change are those in I_0. The equivalence between the two definitions in (14) is trivial.

3.2 Generating automatic corrections

As observed in the introduction, a record can be made to satisfy a violated balance edit by changing one of the variables involved in that edit. In particular, if $i \in \bar{I}_3^{(r)}$ and e_r is currently violated, then the edit becomes satisfied if we change the value of x_i to

$$\tilde{x}_i^{(r)} = \frac{1}{a_{r,i}} \left(s_2^{(r)} - s_1^{(r)} + a_{r,i} x_i \right). \tag{15}$$

Namely, if $i \in I_1^{(r)}$ then this operation changes the value of $s_1^{(r)}$ to

$$\tilde{s}_1^{(r)} = s_1^{(r)} - a_{r,i} x_i + a_{r,i} \tilde{x}_i^{(r)} = s_2^{(r)}, \tag{16}$$

and if $i \in I_2^{(r)}$ then this operation changes the value of $s_2^{(r)}$ to

$$\tilde{s}_2^{(r)} = s_2^{(r)} + a_{r,i} x_i - a_{r,i} \tilde{x}_i^{(r)} = s_1^{(r)}. \tag{17}$$

In both cases, the edit is no longer violated.[1]

For each $i \in I_0$, a list of values $\tilde{x}_i^{(r)}$ can be generated by computing (15) for all $r \in \bar{R}_3^{(i)}$. Next, we check, for each value on the list, whether a simple typing error could have produced the observed value x_i if the true value were $\tilde{x}_i^{(r)}$. This is the case if

$$x_i = f(\tilde{x}_i^{(r)}) \tag{18}$$

for one of the functions introduced in Section 2. If a function can be found such that (18) holds, it seems plausible that a simple typing error has changed the true value $y_i = \tilde{x}_i^{(r)}$ to the observed value x_i. Before drawing any conclusions, however, it is important to examine all other possible corrections. For now, we keep the value $\tilde{x}_i^{(r)}$ on the list. On the other hand, if no function is found such that (18) holds, then $\tilde{x}_i^{(r)}$ is removed from the list, because no simple typing error could have changed this value into the observed value x_i.

After discarding some of the values from the list, it is possible that only an empty list remains. In that case, we do not consider this variable anymore. On the other hand, the reduced list may contain duplicate values, if the same value of x_i can be used to satisfy more than one edit. We denote the unique values that occur on the reduced list by $\tilde{x}_{i,1}, \ldots, \tilde{x}_{i,T_i}$, and we denote the number of times that value $\tilde{x}_{i,t}$ occurs by $\kappa_{i,t}$. If $T_i = 1$, we drop the second index and simply write $\tilde{x}_i$ and κ_i. We remark that $\kappa_{i,t}$ represents

[1] In the case that $|a_{r,i}| > 1$ (which we have not actually encountered in practice at Statistics Netherlands), it is possible that formula (15) yields a non-integer value. As an example, consider a record with $x_2 = 4$ and $x_3 = 11$, where we want to find the value of x_1 such that $2x_1 + x_2 = x_3$ holds. Using (15), we obtain $x_1 = 7/2$. For our present purpose, a non-integer $\tilde{x}_i^{(r)}$ can be immediately discarded, because it is never explained by a simple typing error.

the number of currently violated edits that become satisfied when x_i is changed to $\tilde{x}_{i,t}$. By construction, it holds that $\kappa_{i,t} \geq 1$.

The above procedure is performed for each $i \in I_0$. For each variable, we find a (possibly empty) list of potential changes that can be explained by simple typing errors and that, when considered separately, cause one or more violated edits to become satisfied. The question now remains how to make an optimal selection from these potential changes. Ideally, the optimal selection should return the true values of all variables affected by simple typing errors. Since we do not know the true values, a more pragmatic solution is to select the changes that together lead to a maximal number of satisfied edits. In the simple case that exactly one potential change is found for exactly one variable, the choice is straightforward. If more than one potential change is found and/or if more than one variable can be changed, the choice requires more thought, because clearly, we cannot change the same variable twice and we should not change two variables involved in the same edit. On the other hand, a record might contain several independent typing errors, and we do want to resolve as many of these errors as possible.

The selection problem from the previous paragraph can be formulated as a mathematical optimisation problem:

$$\begin{aligned} &\text{maximise } \textstyle\sum_{i\in I_0}\sum_{t=1}^{T_i} \kappa_{i,t}\delta_{i,t}, \text{ such that:} \\ &\textstyle\sum_{i\in \bar{I}_3^{(r)}\cap I_0}\sum_{t=1}^{T_i} \delta_{i,t} \leq 1, \text{ for } r \in E_1, \\ &\delta_{i,t} \in \{0,1\}, \text{ for } i \in I_0 \text{ and } t \in \{1,\ldots,T_i\}. \end{aligned} \tag{19}$$

The binary variable $\delta_{i,t}$ equals 1 if we choose to replace x_i with the value $\tilde{x}_{i,t}$, and 0 otherwise. Note that the criterion function in (19) counts the number of resolved edit violations. We seek values for $\delta_{i,t}$ that maximise this number, under the inequality constraints in (19). These constraints state that at most one change is allowed for each $i \in I_0$, and that at most one variable may be changed per violated edit. Here, the assumption is used that each variable is involved in at least one edit.

To solve problem (19), we may apply a standard branch and bound algorithm, constructing a binary tree to enumerate all choices of $\delta_{i,t}$. Branches of the binary tree may be pruned if they do not lead to a feasible solution with respect to the inequality constraints. Note that in this case many branches can be pruned because the constraints are quite strict: once we set $\delta_{i,t} = 1$ for a particular (i,t), all other δ-values that occur in the same constraint must be set to zero. This helps to speed up the algorithm.

Once a solution to (19) has been found, the value of x_i is changed to $\tilde{x}_{i,t}$ if $\delta_{i,t} = 1$. If $\delta_{i,t} = 0$ for all $t = 1,\ldots,T_i$, then the value of x_i is not changed. Formally, for each $i \in I_0$ the new value $\hat{x}_i$ is given by

$$\hat{x}_i = \sum_{t=1}^{T_i} \tilde{x}_{i,t}\delta_{i,t} + x_i\left(1 - \sum_{t=1}^{T_i}\delta_{i,t}\right). \tag{20}$$

In the next section, we work out a small-scale example to illustrate the method.

4 Example

Suppose that the unedited data consist of records with $n = 11$ numerical variables that should conform to $m = 5$ balance edits:

$$\begin{cases} e_1: & x_1 + x_2 = x_3 \\ e_2: & x_2 = x_4 \\ e_3: & x_5 + x_6 + x_7 = x_8 \\ e_4: & x_3 + x_8 = x_9 \\ e_5: & x_9 - x_{10} = x_{11} \end{cases} \tag{21}$$

The corresponding partitions (9) and (11) are displayed in two tables in Appendix A.

Throughout this section, we use the following correct but unobserved record $\mathbf{y}$:

y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}
1452	116	1568	116	323	76	12	411	1979	1842	137

This record satisfies all edits (21). Below, we consider four different observed versions of $\mathbf{y}$ that contain simple typing errors.

4.1 A record with one simple typing error

The first record $\mathbf{x}$ we consider has the following observed values:

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	1979	1842	137

Edit e_2 is the only violated edit, because $x_2 = 116$ and $x_4 = 161$. This means that $E_1 = \{2\}$ and $E_2 = \{1,3,4,5\}$. Using (14), we find that $I_0 = \{4\}$; x_4 is the only variable not involved in any satisfied edit. Since x_4 is only involved in edit e_2, formula (15) yields one possible value: $\tilde{x}_4^{(2)} = 116$. From this value, the observed value $x_4 = 161$ can be explained by a simple typing error, namely the interchanging of two adjacent digits in the true value. Formally, $f_{\text{ic}}(116;0) = 161$. Since there is only one potential change to consider in this example, we do not have to set up an optimisation problem, but simply replace $x_4 = 161$ with the new value $\hat{x}_4 = 116$. Comparing the resulting record with $\mathbf{y}$, we see that the true value $y_4 = 116$ has been recovered.

4.2 A record with two simple typing errors

Next, we consider the following observed record:

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

This record violates edits e_2, e_4 and e_5. Thus $E_1 = \{2,4,5\}$, $E_2 = \{1,3\}$ and $I_0 = I_3^{(1)} \cap I_3^{(3)} = \{4,9,10,11\}$; the variables x_4, x_9, x_{10} and x_{11} are only involved in violated edits. Just like in the previous example, we find the potential new value $\tilde{x}_4 = 116$.

Choosing this value only changes the status of edit e_2 from violated to satisfied, so $\kappa_4 = 1$. Variable x_9 is involved in edits e_4 and e_5. According to (15),

$$\begin{aligned} \tilde{x}_9^{(4)} &= -(19979 - 1979 - 19979) = 1979, \qquad \text{and} \\ \tilde{x}_9^{(5)} &= 1979 - 19979 + 19979 = 1979. \end{aligned}$$

so both edits become satisfied by the same choice of $\tilde{x}_9$. Moreover, the observed value can be explained by a simple typing error, since $f_{\mathrm{a}}(1979;3,9) = 19979$. Thus, we find $\tilde{x}_9 = 1979$ with $\kappa_9 = 2$. Variables x_{10} and x_{11} are only involved in edit e_5, and we find:

$$\begin{aligned} \tilde{x}_{10}^{(5)} &= -(1979 - 19979 - 1842) = 19842, \qquad \text{and} \\ \tilde{x}_{11}^{(5)} &= -(1979 - 19979 - 137) = 18137. \end{aligned}$$

Changing 19842 to 1842 can be explained by a simple typing error ($f_{\mathrm{o}}(19842;3) = 1842$), so $\tilde{x}_{10} = 19842$ with $\kappa_{10} = 1$. Changing 18137 to 137 requires multiple typing errors, so we do not consider variable x_{11} anymore.

Since several potential changes have been found, we set up problem (19) to determine the optimal choice. We obtain:

$$\begin{aligned} &\max \{\delta_4 + 2\delta_9 + \delta_{10}\}, \text{ such that:} \\ &\delta_4 \leq 1, \\ &\delta_9 \leq 1, \\ &\delta_9 + \delta_{10} \leq 1, \\ &\delta_4, \delta_9, \delta_{10} \in \{0,1\}. \end{aligned}$$

It is easy to see that the optimal solution is: $\{\delta_4 = 1,\ \delta_9 = 1,\ \delta_{10} = 0\}$. This solution yields the following changes in the observed record: $\hat{x}_4 = 116$ and $\hat{x}_9 = 1979$. The resulting record satisfies all edits and is identical to $\mathbf{y}$.

4.3 A record with multiple errors

Now, suppose that the observed record contains the two simple typing errors from the previous example, as well as a different kind of error:

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	0	0	0	411	19979	1842	137

The only non-violated edit is e_1, so $I_0 = I_3^{(1)} = \{4,5,6,7,8,9,10,11\}$. The reader may verify that we find the same potential changes as before for x_4, x_9, x_{10} and x_{11}. Variable x_5 is only involved in edit e_3. In order to satisfy this edit, the value of x_5 should be changed from 0 to 411. Clearly, this cannot be explained by a simple typing error. The same result holds for x_6 and x_7. Finally, variable x_8 is involved in two edits, and formula (15) yields:

$$\begin{aligned} \tilde{x}_8^{(3)} &= -(411 - 0 - 411) = 0, \qquad \text{and} \\ \tilde{x}_8^{(4)} &= 19979 - 1979 + 411 = 18411. \end{aligned}$$

Neither of these changes can be explained by a simple typing error.

Since the same potential changes are found for this record as for the previous example, the same optimisation problem is constructed and the same optimal solution is found. The resulting record is:

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	116	0	0	0	411	1979	1842	137

All simple typing errors have been successfully removed, but edit e_3 remains violated. A more advanced method is needed to resolve the remaining error, e.g. an implementation of the Fellegi-Holt paradigm. The point to be made here is that the error localisation problem has been substantially simplified for this record, because a number of errors have been resolved by our deductive method.

4.4 Another record with multiple errors

In the previous example, it was possible to correct all simple typing errors, despite the presence of other errors. Unfortunately, this is not always true, as the next example demonstrates. The following observed record has (by now familiar) simple typing errors in x_4 and x_9, and in addition the value of x_8 is reported erroneously:

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	0	19979	1842	137

Again, the only non-violated edit is e_1, so all variables from x_4 to x_{11} should be checked.

The reader may verify that the interchanged digits in x_4 are still found ($\tilde{x}_4 = 116$ with $\kappa_4 = 1$), and that the potential changes in x_5, x_6, x_7, x_8 and x_{11} are not simple typing errors. For x_9, formula (15) yields:

$$\begin{aligned} \tilde{x}_9^{(4)} &= -(19979 - 1568 - 19979) = 1568, \qquad \text{and} \\ \tilde{x}_9^{(5)} &= 1979 - 19979 + 19979 = 1979. \end{aligned}$$

The first value cannot be explained by a simple typing error and is discarded. The second value is the same as before, but in this example it only makes edit e_5 become satisfied: $\tilde{x}_9 = 1979$ and $\kappa_9 = 1$. Finally, we find $\tilde{x}_{10} = 19842$ with $\kappa_{10} = 1$ as before.

This time, the following instance of problem (19) is constructed:

$$\begin{aligned} &\max \{\delta_4 + \delta_9 + \delta_{10}\}, \text{ such that:} \\ &\delta_4 \leq 1, \\ &\delta_9 \leq 1, \\ &\delta_9 + \delta_{10} \leq 1, \\ &\delta_4, \delta_9, \delta_{10} \in \{0, 1\}. \end{aligned}$$

The optimal value of the objective function equals 2, and there are two optimal solutions: $\{\delta_4 = 1, \delta_9 = 1, \delta_{10} = 0\}$ and $\{\delta_4 = 1, \delta_9 = 0, \delta_{10} = 1\}$. Corresponding to these solutions are two corrected versions of the observed record:

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	116	323	76	12	0	1979	1842	137	solution 1
1452	116	1568	116	323	76	12	0	19979	19842	137	solution 2

In this example, we can compare these records with the true record $\mathbf{y}$ and see that the first solution is the best match. In practice of course, the true record is unobserved and there is no way to choose the correct solution. However, it should be noted that $\hat{x}_4$ has the same value in both solutions, so we can safely perform this deductive correction. In general, if problem (19) yields more than one optimal solution, we can still perform deductive corrections for variables that have the same value in all solutions.

5 Refinements

In this section, we describe several improvements to the basic method of Section 3. These refinements were mostly developed while reviewing early results of the practical application discussed in the next section.

5.1 Rounding errors

Quite often, a balance edit of the form $\mathbf{a}'\mathbf{x} = 0$ is violated by a small amount. We call such violations rounding errors if they are small enough, say $|\mathbf{a}'\mathbf{x}| \leq \delta$. For more information on rounding errors and a heuristic method to correct them, see Scholtus (2008). When searching for other types of errors, it is convenient to ignore rounding errors, that is: treat edits that are violated by a small amount as though they were satisfied. With this in mind, we define $\phi(r) = 0$ if

$$-\delta \leq s_1^{(r)} - s_2^{(r)} \leq \delta$$

for the current record, and $\phi(r) = 1$ otherwise. Thus, using this definition in (13), E_2 is the index set of edits that are either satisfied or merely violated because of a rounding error, and E_1 is the index set of edits that are truly violated by the current record.

5.2 Extension to other types of edits

We have assumed until now that only balance edits are specified. In practice, numerical data often also have to satisfy inequalities, such as (2), and conditional edits, such as

$$\text{if } \textit{wages} > 0, \text{ then } \textit{number of employees} > 0. \tag{22}$$

There is an obvious way to extend the method to this more general situation. First, all non-balance edits are ignored and a list of possible corrected values is constructed using formula (15), as before. Now, when reducing the list to $\tilde{x}_{i,1}, \ldots, \tilde{x}_{i,T_i}$, we use an additional criterion: a potential correction should not introduce any new edit violations in the set of inequalities and conditional edits. If a potential correction does lead to new edit violations, it is removed from the list. The rest of the method remains the same.

5.3 Assigning weights

Finally, we suggest a theoretical refinement which has not yet been implemented or tested in practice. Consider the following generalisation of optimisation problem (19):

$$\begin{aligned} &\text{maximise } \textstyle\sum_{i\in I_0}\sum_{t=1}^{T_i} w_{i,t}\delta_{i,t}, \text{ such that:} \\ &\textstyle\sum_{i\in \bar{I}_3^{(r)}\cap I_0}\sum_{t=1}^{T_i}\delta_{i,t} \leq 1, \text{ for } r\in E_1, \\ &\delta_{i,t}\in\{0,1\}, \text{ for } i\in I_0 \text{ and } t\in\{1,\ldots,T_i\}. \end{aligned} \tag{23}$$

Here, weights $w_{i,t}$ are used to incorporate other information into the problem besides $\kappa_{i,t}$, the number of edits that become satisfied by setting $\delta_{i,t}=1$. We suggest taking $w_{i,t}=\kappa_{i,t}g_{i,t}$, where the factor $g_{i,t}$ is based on other information.

For example, weights could be chosen that increase with the number of digits of $\tilde{x}_{i,t}$, since typing errors are more likely to occur in large values, for the simple reason that most people find it "difficult" to handle large numbers. Also, we can assign a zero weight to cases that we do not accept as simple typing errors. For instance, if the observed value $x_i=10$ can be replaced by $\tilde{x}_{i,t}=0$ to satisfy an edit, this is a simple typing error according to the method of Section 3 (adding '1' in the first digit position). We may be reluctant to accept this, if we feel that this is not really an example of a simple typing error. Thus, more convenient results can be obtained by automatically assigning zero weights to certain cases, in particular those cases where either the observed value or the suggested value has only one digit.[2]

We could also assign different weights to values that are explained by different typing errors, if we believe that some typing errors are more likely to occur than others. For instance, if we believe that interchanging two digits occurs more often in practice than adding a digit, a value $\tilde{x}_{i,t}$ that explains the observed value by interchanging two digits can be given a higher weight than a value that explains the observed value by adding a digit. This is mainly a theoretical possibility, because to quantify the difference we need empirical evidence on the occurrence of simple typing errors under real-world conditions. Collecting and analysing such information is probably too costly in practice.

6 Practical application

A prototype implementation of the algorithm was written in the R programming language. This implementation searches for all four simple typing errors introduced in Section 2 simultaneously. Rounding errors are taken into account as discussed in Section 5.1. If more than one optimal solution to (19) exists, the algorithm returns the corrections that are common to all optimal solutions, as we did in the example of Section 4.4.

The prototype was tested using survey data from the Dutch wholesale structural business statistics of 2007. The data file contains 4,381 records with 97 variables each.

[2] In our practical application (see Section 6), the same goal was achieved more bluntly by dropping all suggested corrections with either $x_i<10$ or $\tilde{x}_{i,t}<10$.

Each record should conform to 123 hard edits[3], including 19 balance edits. The original captured data cause $2,026$ violations of balance edits. If rounding errors (with $\delta = 2$) are ignored, $1,758$ balance edit violations remain.

Applying our method revealed 152 simple typing errors, occurring in 143 records. Table 1 shows the number of hits for each type of error. By correcting these simple typing errors, 195 violations of balance edits are removed. Thus, nearly ten percent of all balance edit violations in this data set can be explained by one of the four simple typing errors of Section 2.

Table 1: Number of simple typing errors in the wholesale data

type of error	number of corrections
interchanged digits	14
added digit	11
omitted digit	15
replaced digit	112
total	152

Table 2: Results on the wholesale data by questionnaire type

medium	number of records	number of corrections	violated balance edits (before)	violated balance edits (after)
paper	570	96	1,069	930
electronic	3,811	56	957	901
total	4,381	152	2,026	1,831

The collection of survey data for the Dutch structural business statistics of 2007 was conducted partly using a paper questionnaire and partly using an electronic questionnaire. To obtain a digital file of all survey data, information from paper questionnaires needs to be keyed in at the statistical office. Since this introduces an extra source of simple typing errors, we expect that such errors occur more frequently in data from paper questionnaires than data from electronic questionnaires. Moreover, most of the balance edits were built into the electronic questionnaire, because subtotals were automatically computed from the corresponding items; cf. Giesen (2007). Therefore, we

[3] An edit is called a hard edit (or fatal edit) if it identifies errors with certainty. An edit which can sometimes be violated by correct values is called a soft edit (or query edit).

expect that few balance edit violations occur in data that were collected electronically.

Table 2 shows the results of the application on wholesale data, differentiated by questionnaire medium. These results are in line with our expectations: relatively speaking, both the number of violated balance edits and the number of corrected simple typing errors are much higher among data from the paper questionnaire than data from the electronic questionnaire.

7 Conclusion

In this paper, we have described a method to correct simple typing errors in numerical data, such as interchanged digits, automatically. The method was developed with the microdata of the Dutch structural business statistics in mind, but it can be used in any application where numerical data should conform to balance edits.

The list of simple typing errors given in Section 2 is not exhaustive. The method from this paper can be used for the automatic detection of any error that only affects one variable at a time and has a distinct, easily recognisable effect on the value of that variable. In itself, the detection of such an error is almost trivial. The only complication derives from the fact that the data have to satisfy many inter-related edits, so that if a variable is changed to satisfy one edit, this may cause a new violation of another edit. Our method describes a way to take all edits into account simultaneously.

References

De Jong, A. (2002), 'Uni-Edit: Standardized Processing of Structural Business Statistics in The Netherlands'. Paper presented at the UN/ECE Work Session on Statistical Data Editing, 27-29 May 2002, Helsinki, Finland.

Fellegi, I. P. and Holt, D. (1976), 'A Systematic Approach to Automatic Edit and Imputation', *Journal of the American Statistical Association* **71**, pp. 17–35.

Giesen, D. (2007), 'Does Mode Matter? First Results of the Comparison of the Response Burden and Data Quality of a Paper Business Survey and an Electronic Business Survey'. Paper presented at QUEST 2007, 24-26 April 2007, Ottawa, Canada.

Scholtus, S. (2008), 'Algorithms for Correcting Some Obvious Inconsistencies and Rounding Errors in Business Survey Data'. Discussion Paper 08-015, Statistics Netherlands, The Hague.

Van de Pol, F., Bakker, F. and De Waal, T. (1997), 'On Principles for Automatic Editing of Numerical Data with Equality Checks'. Report 7141-97-RSM, Statistics Netherlands, Voorburg.

Appendix A Tables

The following tables display the partitions $I_1^{(r)}, I_2^{(r)}, I_3^{(r)}$ and $R_1^{(i)}, R_2^{(i)}, R_3^{(i)}$ for the example from Section 4.

r	$I_1^{(r)}$	$I_2^{(r)}$	$I_3^{(r)}$
1	$\{1,2\}$	$\{3\}$	$\{4,5,6,7,8,9,10,11\}$
2	$\{2\}$	$\{4\}$	$\{1,3,5,6,7,8,9,10,11\}$
3	$\{5,6,7\}$	$\{8\}$	$\{1,2,3,4,9,10,11\}$
4	$\{3,8\}$	$\{9\}$	$\{1,2,4,5,6,7,10,11\}$
5	$\{9\}$	$\{10,11\}$	$\{1,2,3,4,5,6,7,8\}$

i	$R_1^{(i)}$	$R_2^{(i)}$	$R_3^{(i)}$
1	$\{1\}$	$\emptyset$	$\{2,3,4,5\}$
2	$\{1,2\}$	$\emptyset$	$\{3,4,5\}$
3	$\{4\}$	$\{1\}$	$\{2,3,5\}$
4	$\emptyset$	$\{2\}$	$\{1,3,4,5\}$
5	$\{3\}$	$\emptyset$	$\{1,2,4,5\}$
6	$\{3\}$	$\emptyset$	$\{1,2,4,5\}$
7	$\{3\}$	$\emptyset$	$\{1,2,4,5\}$
8	$\{4\}$	$\{3\}$	$\{1,2,5\}$
9	$\{5\}$	$\{4\}$	$\{1,2,3\}$
10	$\emptyset$	$\{5\}$	$\{1,2,3,4\}$
11	$\emptyset$	$\{5\}$	$\{1,2,3,4\}$