Macro-Editing with Blaise III

Jelke Bethlehem and Lon Hofman, Statistics Netherlands

1. Introduction

Sample surveys are carried out to collect information about a specific population. Results are presented in the form of statistics, i.e. estimates of unknown population characteristics. The statistics will rarely correspond exactly to the unknown values of the population characteristics. Every survey operation is affected by errors, and the magnitude of these errors affects the accuracy of the results.

Generally, two broad categories of errors are distinguished: sampling errors and nonsampling errors. Sampling errors are due to the fact that only a sample is observed and not the entire population. Every replication of a survey experiment would result in a different sample, and therefore in different estimates. Sampling errors will not occur in a census (a complete enumeration of the population). The survey researcher is in control of the sampling error. By making a proper sampling design, and selecting a sufficiently large sample, unbiased estimates can be obtained with small standard errors. Of course, the available financial budget may impose restrictions on the size of the sample.

Nonsampling errors relate to all survey errors originating from sources other than the use of a sampling mechanism. Such errors will also occur in a complete enumeration of the population. Examples of nonsampling errors are errors due to nonresponse, measurement errors (e.g. wrong answers), coverage errors, and processing errors. Nonsampling errors are difficult to control. The best way to deal with nonsampling errors is to take preventive measures at the design stage. One can think of well-considered selection of the sampling frame, careful design of the questionnaire, adequate training of interviewers, and thorough pilot surveys. Nevertheless, many causes of nonsampling errors are inherent in the survey process, and as such are virtually impossible to avoid.

Nonsampling errors have an impact on the quality of survey results. To avoid publication of inaccurate statistics, the collected data must undergo extensive treatment. This treatment is called the data editing process. Aim of data editing is to detect and remove errors in the data. Traditionally, data editing was a manual process, carried out by subject-matter experts. It was costly and time-consuming, and not very effective. The introduction of computers for data editing improved the situation dramatically. Particularly, systems like Blaise provide a powerful and user-friendly environment for extensive error checking, and assisting the subject-matter experts in correcting these errors.

Many systems for data editing are micro-editing systems. *Micro-editing* means that each questionnaire form is processed separately. Checks compare the answers within one form, and do not take into account the results on other forms. The micro-editing approach is able to detect many errors, but not those relating to the distribution of answers over the forms. For example, it is not possible to detect outliers, i.e. answers that are within the range of valid answers, but which differ so much from other answers that probably something is wrong. Such errors can only be detected if all answers to the relevant questions are available.

The data editing approach dealing with distributional aspects of the answers to the questions is called *macro-editing*. Typically, macro-editing is a file-oriented approach, whereas micro-editing is a form-oriented approach. Therefore, macro-editing can not be carried out during data collection, but only after all (or a large part of the) data has been collected.

The Blaise system was originally designed as a form-oriented system. The idea was to integrate a number of activities required to collect, enter, and edit survey data. The form-oriented approach of Blaise makes the system very suitable for micro-editing. This papers describes how Blaise can also be used for macro-editing activities. Using Blaise tools like the DEP and Manipula, we describe simple ways to implement some forms of macro-editing procedures.

Section 2 describes macro-editing in some more detail. Particularly, it is explained why users should sometimes not concentrate too much on micro-editing, and pay more attention to macro editing. Sections 3 and 4 discuss some techniques for macro-editing. Section 5 describes how some of these techniques can be implemented in the current version of Blaise. Section 6 describes a prototype of a special data viewer that allows for graphical macro-editing.

This paper should not be seen as a document containing the specifications of modules that will be in the next release of Blaise. It is discussion paper proposing a possible future way to extend the system. The authors appreciate any comments on the contents of the paper.

2. Why macro-editing?

Traditionally, most survey data is collected by means of paper questionnaire forms. In the process of asking the questions, writing down the answers, and entering the information into the computer, many thing can go wrong. Respondents may not understand a question, and therefore give a wrong answer. They may understand the question, but do not want to give the right answer (e.g. for sensitive questions). Furthermore, interviewers can mistakes in writing down the answers, and data entry operators can make errors when entering the data. Consequently, the information in the data file does always not reflect the real situation being investigated. Analysis of such data may lead to wrong conclusions. Therefore, it is vital to detect and correct any errors. This process is called editing.

The traditional approach to editing is a manual form of micro-editing. After collection of the forms, subject-matter specialists check the forms for completeness. If necessary and possible, skipped questions are answered, and obvious errors are corrected on the forms. Sometimes, the forms are manually copied to a new form to allow for the subsequent step of data entry. Next, the forms are transferred to the data entry department. Data typists enter the data in the computer at high speed without much error checking. Usually, only some simple range checks are carried out to detect data entry mistakes.

After data entry, an error detection program is run. Detected errors are printed on lists. The lists with errors are sent to the subject-matter experts. They investigate the error messages, consult corresponding forms, and correct errors on the lists where possible. Lists with corrections are sent back to the data entry department, and data typists enter the corrections, after which corrected records and already accepted records are merged.

Usually, the cycle of batch-wise error detection and manual correction is repeated a number of times, until the number of rejected records is sufficiently small.

The data editing procedure described here, has it limitations. Since different departments and different computer systems are involved in a cyclic process, it is time-consuming. Furthermore, the manual activities by the subject-matter experts are costly, and not very effective.

To improve the efficiency of the statistical production process, and the quality of the produced statistical information, Statistics Netherlands developed different approaches to data collecting and data editing.

CADI (Computer-assisted data input) was applied in economic surveys. The idea was to improve the handling of paper questionnaire forms by integrating data entry and data editing tasks. The traditional batch-oriented data editing activities, in which the complete data set was processed as a whole, was replaced by a record-oriented process in which records (forms) were completely dealt with one at a time. Basically, CADI is a form of micro-editing. CADI was implemented in version 1 of Blaise. It is used in two ways.

Subject-matter specialists take care of both data entry and data editing, thus avoiding the cycle through different departments.

Data typists use the CADI system to enter data without much error checking. After completion, the CADI system checks in a batch run all records, and flags the rejected ones. Then subject-matter specialist handles the rejected records one by one, and correct the detected errors (also with a CADI system).

Computer-assisted interviewing (CAI) techniques are applied in social surveys. The paper questionnaire is replaced by a computer program containing the questions to be asked. The computer takes control of the interviewing process.

The computer program determines the route through the questionnaire. It determines which question is to be asked next, and displays that question on the screen. Such a decision may depend on the answers to previous questions. Hence it relieves the interviewer of the task of taking care of the correct route through the questionnaire. As a result, it is not possible anymore to make route errors.

The computer program also checks the answers to the questions which are entered. Range checks are carried out immediately after entry, and consistency checks after entry of all relevant answers. If an error is detected, the program gives a warning, and one or more of the answers concerned can be modified. The program will not proceed to the next question until all detected errors have been corrected.

The CAI approach also implements a form of micro-editing. The big difference with the CADI approach is that micro-editing is moved from the statistical office to the field. All micro-editing is taken care off during the interview, so that the statistical office only receives ‘clean’ (accepted) forms. Almost no editing activities are left for the office. CAI was implemented in version 2 of Blaise.

Systems like Blaise are very powerful instruments for micro-editing. They have simple facilities to specify a large number of checks. Checks may also analyse complex relationships between many questions. Application of these micro-editing systems has proved to be successful. They are capable of detecting and correcting many errors, and therefore they improve data quality. However, micro-editing also has some drawbacks.

The first point to be mentioned is the risk of ‘over-editing’. With systems like Blaise it easy to implement almost any check one can think of. On the one hand, this seems to be a nice feature, based on the idea that the more checks are carried out, the more errors will be corrected. However, there are also disadvantages:

- It is possible to implement checks that contradict one another. This will cause all records to be rejected.
- It is possible to implement redundant checks. In such a situation, one problem may result in many error messages. This will make the work of the subject-matters workers or the interviewers very difficult.
- It is possible to implement checks that detects problems that almost have no impact on the quality of the published statistical information. Such checks generate a lot of work that does not pay off.

The second point to be made with respect to micro-editing, is that this approach is not capable of detecting all problems. One example of such a problem is outliers. An outlier refers to a value of a variable (answer to a question) that is within the defined range of valid values, but is very unlikely when compared with the distribution of all valid values. An outlier can only be detected of the distribution of all values is available, and for that one needs the whole file with records.

Macro-editing offers a solution to some of the problems of micro-editing. Particularly, it can deal with editing relating to the distributional aspects. Like in micro-editing, checks can take all kinds of forms in macro-editing. Here, two forms are summarised. The first form is sometimes called the ‘aggregation method’. See also Granquist (1990). The idea is to divide the data file in groups, and to compute aggregate statistics for each group. The distribution of the values is analysed (for example with the techniques of section 3), and only if an unusual value is observed, a micro-editing procedure is applied to the individual records in the group. The advantage of this form of editing is that it concentrates on detecting errors that have an impact on the final results of the survey. No superfluous micro-editing activities are carried on records in groups that do not produce unusual values at the aggregate level.

A second form of macro-editing could be called the ‘distribution method’. The available data is used to characterise the distribution of the variables. Then, all individual values are compared with the distribution. Typically, measures of location and spread are computed. Records containing values that could be considered uncommon (given the distribution) are candidates for further inspection and possible editing.

Both forms of macro-editing involve a check component that requires the data file to be available, and a correction component requiring the individual records.

The rest of this paper concentrates on the 'distribution method'. Several techniques to characterise the distribution of one ore more variables are discussed in the subsequent sections. Some of these techniques can implemented in the Blaise system. This is described in sections 5 and 6.

3. Macro-editing techniques for one variable

Many macro-editing techniques analyse the behaviour of a single observation in the distribution of all observations. We will discuss some of these techniques. We restrict ourselves to the analysis of quantitative variables. Quantitative variables measure a size, amount, or value. Typically, it is possible to compute quantities like sums and averages. We will not discuss qualitative variables. These variables divide the observations in groups. Values are group labels. It is not useful to carry out arithmetic operations on the values. Furthermore, we will restrict ourselves to the analysis of a single variable, and to the analysis of the relationship between two variables.

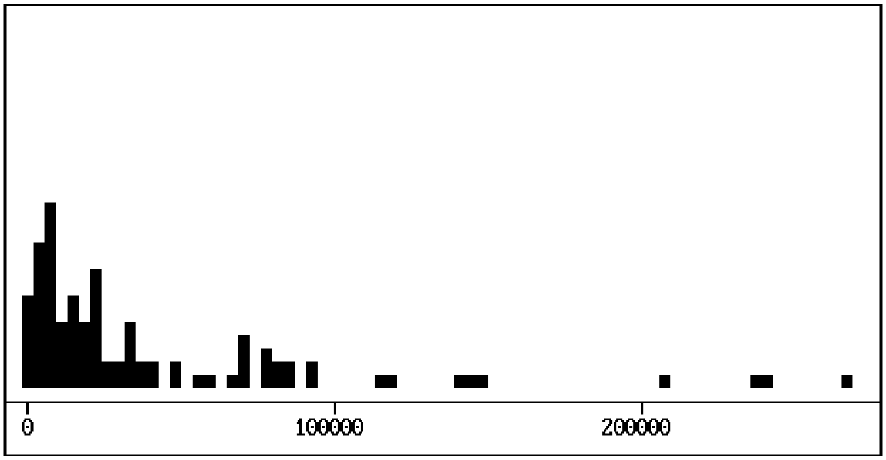
There is an area in statistics providing all kinds of techniques for analysing the distribution of variables, and that is exploratory data analysis (EDA). See for example Tukey (1977). Many of the techniques found here can be applied in macro-editing. Here we concentrate on two groups of techniques. The first group analyses the distribution of a single variable, and concentrates on detecting outliers. The second group analyses the relationship between two variables, and tries to find records showing a different relationship. All techniques discussed can be seen as special cases of the 'distribution method' of macro-editing.

Advocates of EDA stress the importance of the use of graphical techniques. These techniques provide much more insight in the behaviour of variables than numerical techniques do. This also applies to macro-editing. Graphs of the distribution of the data show a lot of information, and are capable of showing unexpected properties that would not have been discovered if just numerical quantities were computed.

In this section, we describe some techniques for the analysis of the distribution of a single quantitative variable. These techniques characterise the distribution by a measure of location, and a measure of spread around this location. They also attempt to detect values with a different behaviour. This special values are called 'outliers'. A typical macro-editing application will concentrate on the analysis and possible treatment of these outliers.

A technique that displays the distribution in its most elementary form, is the *one-way scatterplot*. In such a plot each individual value of the variable is displayed on a horizontal scale. Figure 3.1 shows an example.

Figure 3.1. One-way scatterplot of the manure production



The variable displayed in the plot is the yearly manure production by farms in 98 municipalities in the Dutch province of Zuid-Holland. Each small square represents the manure production within one municipality. Where rounded values coincide, the corresponding squares are stacked. A one-way scatter plot can be used to analyse the following properties of the distribution:

- *Outliers*. Outliers appear as single squares that are far apart from the rest of the observations. Outliers require further analysis, since the may indicate wrong values;

- *Grouping*. Grouping means that values are not more or less evenly spread over the whole domain of possible answers, but instead appear to be concentrated in separate groups. Grouping may be an indication the observations originate from differently behaving subpopulations, and therefore it might be more appropriate to analyse these groups separately;

- *Concentration*. Only if the values concentrate around a certain location, it is acceptable to use this location to characterise the distribution;

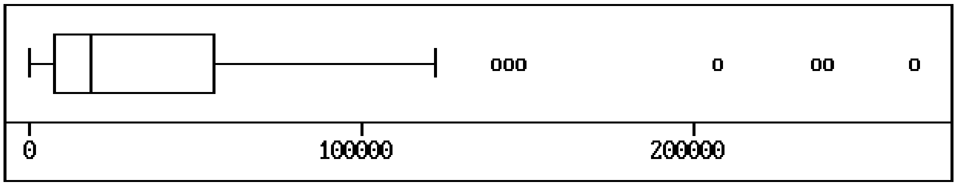
- *Symmetry*. Only if the distribution is symmetric and concentrates around one location, it is acceptable to use numerical techniques assuming normality.

If we look at our example, we see four values at the right-hand side of the distribution that might be considered outliers. There is no grouping of observations. The distribution is far from symmetric, making it very difficult to use numerical quantities like means and standard deviations to characterise the distribution.

For asymmetric distributions, one might consider a transformation. If the transformed distribution resembles the normal one, more techniques are available for analysing the distribution. In case of skew distributions like the one in the example, taking logarithms or square-roots might help.

A next technique to study the distribution the values of a single variable is *the box-and-whisker plot*. The box-and-whisker plot displays the shape of the distribution in a schematic way, without showing all individual values. Figure 3.2 contains an example of a box-and-whisker plot.

Figure 3.2. The box-and-whisker plot of the manure production



The rectangular box represents the middle part of the distribution. It extends from the *first quartile* to the *third quartile*. The line within the box is the *median* (the second quartile). The whiskers connect the box to the so-called *adjacent values*. The *upper adjacent value* is defined to be the largest value less than or equal to the third quartile plus 1.5 times the length of the box. Likewise, the *lower adjacent value* is smallest value greater than or equal to the first quartile minus 1.5 times the length of the box. Any value falling outside the interval between the two adjacent values is considered to be an outlier, and therefore plotted as an individual point. The following aspects of the box-and-whisker plot are worth taking a look at:

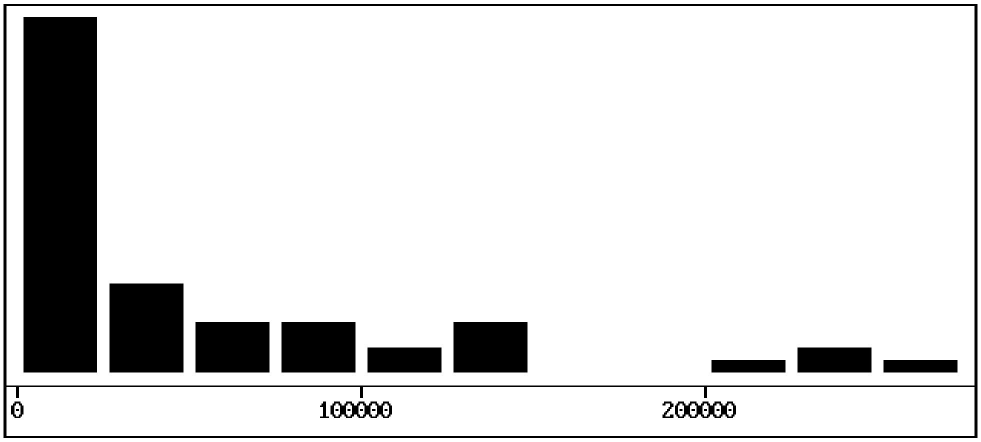
- *Outliers*. Outliers appear as single points. These points require further analysis, since the may indicate wrong values;

- *Symmetry*. If the box-and-whisker plot is symmetric, the underlying distribution is symmetric. Only if the distribution is symmetric and concentrates around one location, it is acceptable to use numerical techniques assuming normality;

- *Length of whiskers*. If the length of the whiskers is much shorter or much longer than the 1.5 times the length of the box, this indicates a deviation from normality. One should be careful in using numerical techniques that assume normality of the underlying distribution.

Finally, a more traditional way of portraying the distribution is the *histogram*. The domain of possible values is divided into a number of intervals of equal length. The number of points in each interval is counted, and these counts are represented as bars with lengths proportional to the counts. Figure 3.3 shows an example of a histogram.

Figure 3.3. Histogram of the manure production



The choice of the number of intervals is rather critical. Too few intervals will cause many details to be hidden, and too many intervals will cause too much focus on the details. An often used rule of thumb is to take the number of interval approximately equal to the square root of the number of observations, with a minimum of 5 and a maximum of 20 intervals.

The histogram is mainly useful for studying the symmetry and concentration of the distribution. In some cases this graph might also be able to detect grouping of values. The histogram is not the proper instrument for detecting outliers.

There are also numerical ways to characterise the distribution, and to search for outliers. One of the most frequently used techniques is based on the mean and variance of the observations. Suppose we have N values X_1, X_2, ..., X_N. Then the *mean* is defined by

$$\overline{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$$

and the *variance* is equal to

$$S^2 = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \overline{X})^2 \ .$$

The *standard deviation* S is defined as the square root of the variance. If the underlying distribution is normal then approximately 95% of the values must lie in the interval

$$(\overline{X} - 2S; \overline{X} + 2S),$$

and approximately 97% in the interval

$$(\overline{X} - 3S; \overline{X} + 3S).$$

Outliers can now be defined as values outside one of these intervals. This simple technique has two important drawbacks:

1) The assumptions are only satisfied if the underlying distribution is approximately normal. So, this technique should only be used if a graphical technique has shown that this model assumption is not unrealistic.

2) The technique is very sensitive to outliers. A single outlier can have a large effect on the values of mean and variance, and therefore even hide the outlier. Also here the message is that the technique should only be used after graphical techniques have shown no dramatical deviations from normality.

Less sensitive techniques are based on the median and the quartiles of the distribution. For example the box-and-whisker plot could be applied in a numerical way. Values smaller than the lower adjacent value or larger than the upper adjacent value would then be marked as outliers.

4. Macro-editing techniques for two variables

Now we discuss some techniques to study the simultaneous distribution of two quantitative variables. Also here we stress the importance of first taking a look at plots before relying on numerical techniques.

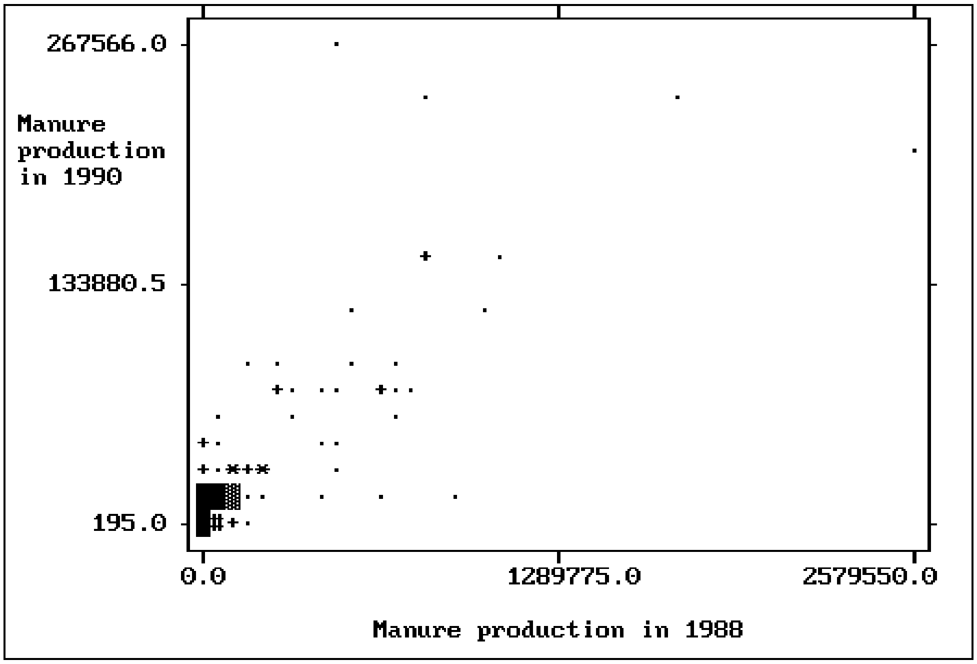
The obvious technique for displaying the relationship between two variables is the *two-dimensional scatterplot*. Suppose, we have two variables X and Y. For all N cases in the sample we have the pairs of values (X_1,Y_1), (X_2,Y_2), ..., (X_N,Y_N). The scatterplot displays each pair (X_i,Y_i) as a point with co-ordinates X_i and Y_i. So, the scatterplot will contain N points, each representing a case. Here are some patterns to look for in a scatterplot:

- *Outliers*. Outliers appear as single points that are far apart from the rest of the observations. Outliers require further analysis, since they may indicate wrong values;

- *Grouping*. Grouping means that points are not more or less evenly spread over the whole domain of possible answers, but instead appear to be concentrated in separate groups. Grouping may be an indication the observations originate from differently behaving subpopulations, and therefore it might be more appropriate to analyse these groups separately;

If clear patterns and structures can be distinguished in the plot, this indicates a certain relationship. The simples form of a relationship is that of a linear relationship. In this case, all points will (approximately) lie on a straight line. If such a relationship seems to hold, it is important to look for points not following the pattern. These point may indicate errors in the data. Figure 4.1 shows an example of a two-dimensional scatterplot.

Figure 4.1. Example of a two-dimensional scatterplot



The variable on the X-axis is the yearly manure production in 1988 by farms in the municipalities in the Dutch province of Zuid-Holland. The variable on the Y-axis is the same variable, but measured in the year 1990. The plot seems to indicate a linear relationship between the two variables, but there are also some points that do not conform to this pattern.

If the assumption of linearity is not unreasonable, it can be summarised in the form of a regression line. The formula for this regression line is

$$Y = a + bX$$

where a and b are coefficients that have to be computed using the available data. For the best line, i.e. the line that follows the pattern most closely, the value of b is equal to

$$b = \frac{\sum_{i=1}^{N}(X_i - \overline{X})(Y_i - \overline{Y})}{\sum_{i=1}^{N}(X_i - \overline{X})^2},$$

and a is equal to

$$a = \overline{Y} - b\overline{X}$$

If the assumption of a linear relationship is correct, the regression line can be used to detect outliers. First, define the *residual* R_i for case i by

$$R_i = Y_i - a - bX_i.$$

The residual is equal to the difference between the value Y_i and the prediction a-bX_i for Y_i based on the regression line. It is the (vertical) distance between the point and the regression. Large residuals indicate outliers. It can be shown that the *studentized residual* T_i, defined by

$$T_i = \frac{R_i}{S(T_i)},$$

where $S(T_i)$ is equal to

$$S(T_i) = \sqrt{1 - \frac{1}{n} - \frac{(\overline{X} - X_i)^2}{\sum_{k=1}^{N}(X_k - \overline{X})^2}},$$

has an approximately normal distribution. Therefore, studentized residuals with values outside the interval (-2;2) or (-3;3) can be considered outliers. These cases need further investigation.

The computation of the coefficients a and b of the regression line is sensitive to outliers. A few big outliers can have a considerable effect on the shape of the regression line. So, be careful to look at the scatterplot before computing regression lines.

There are ways to compute regression lines that are less sensitive to outliers. One technique is iteratively reweighted regression. A weighted regression is repeated a number of times. In each regression, weights are assigned to cases. These weights are determined by the residuals of the previous regression. The larger the residuals, the smaller the weights. For more details, see e.g. Chambers et al. (1983).

5. Simple macro-editing with Blaise and Manipula

There are several ways to implement macro-editing in Blaise. In this paper we want to present two such approaches. The first approach is described in this section. It is a simple approach that does not take full advantage of the graphical possibilities for macro-editing. This approach can be implement using available tools like Blaise and Manipula. The second approach requires an add-on to the Blaise system, but provides a more effective environment for macro-editing. This approach is described in the next section.

We start with macro-editing a single variable, and concentrate on the detection and treatment of outliers. To be able to do that a number of quantities must be computed, and inserted in the data model. In this approach, we go through the following steps:

1) Enter data with the DEP, using CAPI, CATI, or CASI, depending on the survey at hand. Reserve fields in the data model to hold aggregate statistics required for macro-editing;

2) After all data has been collected, use Manipula to compute aggregate statistics, import these statistics in the Blaise data file, and carry out a check on every record;

3) Edit the rejected forms with the DEP in CADI mode.

The traditional approach to finding outliers is based on the prediction interval described in section 3. To compute this interval we need the mean and the standard deviation.

We will describe an example of this approach using an example based on artificial data. It relates to manure production by farms in municipalities in the Dutch province of Zuid-Holland. For two years (1988 and 1990), data has been collected on three variables: area of grass, number of cattle, and manure production. Figure 5.1 contains the Blaise metadata specification for these data.

The data model contains three blocks: one block with information to identify the municipality, one block with the agricultural variables, and one block dealing with macro-editing. The second block is used twice, once for the year 1988, and once for the year 1990. Note that this model is used for reasons of simplicity. It is not a realistic data model. In a more practical situation the data on 1988 would probably be imported, and the data on 1990 would be asked.

Note that this model can be used in two ways. In data entry mode, the DEP is used as a program for computer-assisted interviewing. The macro-editing block is skipped. In data editing mode, the checks in the macro-editing block are carried out.

Once all data has been collected in data editing mode, the aggregate statistics must be computed, imported in the Blaise data file. Then the forms have to be checked. Figure 5.2 contains a Manipula setup that takes care of these activities. The target variable for this macro-editing operation is the manure production in 1990.

The Blaise data file is an update file. This means that in the same Manipula session information is read from and written to the file. The setting *AUTOREAD = NO* implies that Manipula does not automatically reads records from the data file, but only when it is instructed to do so.

The description of the Blaise data file contains the setting *CHECKRULES.* The consequence of this setting is that every processed record is checked. The checks carried out are those defined in the rules section of the Blaise data model *Manure1.*

```
DATAMODEL ManureProduction

TYPE
   TReal    = REAL[12, 3]
   TInteger = 0..99999999

BLOCK TMunicipality
   FIELDS
      MunCode "Municipality code": 0..999
      MunName "Municipality name": STRING[26]
   RULES
      MunCode MunName
ENDBLOCK

BLOCK TData
   PARAMETERS
      Year: INTEGER
   FIELDS
      Grass   "Area of grassland in ^Year": TInteger
      Cattle  "Number of cattle in ^Year" : TInteger
      Manure  "Manure production in ^Year": TInteger
   RULES
      Grass Cattle Manure
ENDBLOCK

BLOCK TMacroEdit
   PARAMETERS
      FieldValue: INTEGER
      FieldName : STRING
   LOCALS
      LBound, UBound: Real
   FIELDS
      Mean  "Mean of ^FieldName"              : TReal
      StDev "Standard deviation of ^FieldName": TReal
   RULES
      Mean.SHOW
      StDev.SHOW
      LBound:= ROUND(Mean - 2 * StDev)
      UBound:= ROUND(Mean + 2 * StDev)
      (FieldValue > LBound) and (FieldValue < UBound)
      "Value of ^FieldValue outside the interval (^LBound ; ^UBound)!"
ENDBLOCK

FIELDS
   Municipality: TMunicipality
   ThisYear    : TData
   LastYear    : TData
```

Figure 5.1. The first Blaise metadata specification

The auxfields section of the Manipula setup contains a block definition. The definition of this block is similar to that in the Blaise data model. Such use of similar block makes it easier to specify manipulations. For example, the manipulate section contains the assignment *UpFile.MacroEdit:= MEdit.* This causes the values of all variables in the block *MEdit* to be copied to the corresponding fields in the block *MacroEdit* of the Blaise data model.

Figure 5.2. Manipula setup for the mean and standard deviation

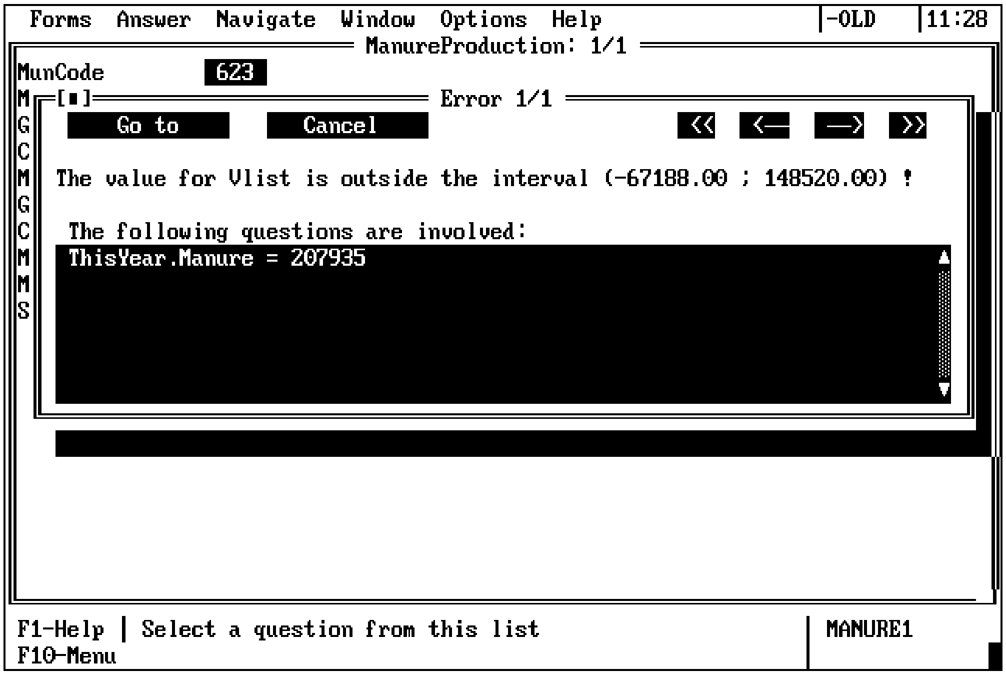
```
SETTINGS
   AUTOREAD = NO

USES
   BlaiseData 'Manure1'

UPDATEFILE
   UpFile: BlaiseData ('Manure1', BLAISE3)
SETTINGS
   CHECKRULES = YES

AUXFIELDS
   TYPE
     TReal = REAL[12, 3]

   BLOCK TMacroEdit
      FIELDS
         Mean : TReal
         StDev: TReal
   ENDBLOCK

FIELDS
   MEdit  : TMacroEdit
   Sx, Sxx: REAL
   RecNum : INTEGER

MANIPULATE
   FOR RecNum:= 1 TO UpFile.RECORDCOUNT DO
       InFile.READNEXT
       Sx := Sx  + ThisYear.Manure
       Sxx:= Sxx + SQR(ThisYear.Manure)
   ENDDO

   Sx:= Sx / MEdit.UpFile.RECORDCOUNT
   MEdit.Mean:= Sx
   MEdit.StDev:= SQRT((Sxx - UpFile.RECORDCOUNT* SQR(Sx)) /
                 UpFile.RECORDCOUNT)

   UpFile.RESET
   FOR RecNum:= 1 TO UpFile.RECORDCOUNT DO
       UpFile.READNEXT
       UpFile.MacroEdit:= MEdit
       UpFile.WRITE
   ENDDO

   READY
```

The manipulate section contains three series of statements. The first series reads the Blaise data file, and computes the sum and the sums of the squares of the field *Manure*. Note the use of the statement *UpFile.READNEXT* to instruct Manipula to read the next record. The second series of statements computes the mean and standard deviation, and assigns these values the corresponding variables in the block *MEdit*. The third series of statements resets the Blaise data file, and changes each record by reading it, copying the contents of the block MEdit to the block MacroEdit, and writing the record back to the file.

Now all information is available to carry out a macro-editing operation. By activating the Data Viewer from the Blaise Control Centre, we can take a look at the data. Note that is possible in the current version of Blaise to see the values of the status variable. This means that we can see whether forms are clean, suspect, or wrong.

We activate the DEP and run the program in in data editing mode (through the option Data entry mode in the Options menu). All records with a value of *ThisYear.Manure* outside the interval *(Mean ± 2 StDev)* have an error attached to the field *Manure*. Figure 5.4 shows an example of an error message.

Figure 5.3. Example of an error message



MunCode 623

The value for Vlist is outside the interval (-67188.00 ; 148520.00) !

The following questions are involved:

ThisYear.Manure = 207935

F1-Help | Select a question from this list

F10-Menu

MANURE1

Apparently, the manure production in the Municipality of Vlist is so high that is marked as an outlier. Note that the interval has a negative lowerbound, although all manure values are greater than or equal to zero. This rather strange lowerbound is caused by the fact that mean and standard deviation are used in a situation where the underlying distribution does not resemble the normal one. In fact, the distribution is very skew, with a lot of small values, and a few large values (see figure 3.1). In this case it would have been better to carry out a transformation on the distribution before attempting to detect outliers.

Note that mean and variance of the variable are computed in the Manipula setup, and the bounds of the interval in the data model. Of course, the bounds could also have been computed in the Manipula setup, but then the values of mean and variance would not have been available (for other purposes) in the data model.

A similar approach could have been used to detect outliers based on the more robust approach of the adjacent values in the box-and-whisker plot. Of course, that would need a different version of the Manipula setup and the macro-editing block in the data model.

We will now show how the analysis of two variables can be implemented using Blaise and Manipula. We want to compare the manure production in 1990 with that of 1988, and studentized residuals outside the interval (-2 ; 2) must be flagged as outliers.

Figure 5.4 contains a part of the Blaise metadata specification for this example. The only difference with the Blaise specification of figure 5.1 is the block with the macro-editing specification. Figure 5.4 contains that block for our two-variable example.

Figure 5.4. The macro-editing block for the two-variable analysis

```
BLOCK TMacroEdit
   PARAMETERS
      XFieldValue: INTEGER
      XFieldName : STRING
      YFieldValue: INTEGER
      YFieldName : STRING

   FIELDS
      N     "Number of records"                : 0..999
      Mean  "Mean of ^XFieldName"              : TReal
      StDev "Standard deviation of ^XFieldName": TReal
      A     "Regression coefficient A"         : TReal
      B     "Regression coefficient B"         : TReal
      RDev  "Standard error of residuals"      : TReal

   AUXFIELDS
      StudRes "Studentized residual": TReal

   RULES
      A.SHOW
      B.SHOW
      StudRes:= (YFieldValue - A - B * XFieldValue) /
      SQRT(1.0 - 1/N - SQR(Mean - XFieldValue) / (SQR(StDev) * N)) / RDev

      (StudRes > -2) AND (StudRes < 2) INVOLVING(YFieldValue)
      "The studentized residual for ^YFieldValue is ^StudRes.
      @/This value is outside the interval (-2 ; 2) !"

ENDBLOCK
```

The manure production in 1990 will play the role of the Y-variable inour example, and the manure production in 1988 will be the X-variable. Given the regression statistics, the rules section of the block *TMacroEdit* computes the studentized residual.

After all data has been collected, the regression information can be computed with Manipula. Figure 5.5 contains a Manipula setup for this purpose.

The setup has the same structure as the one for the one-variables case. The Blaise data file is an update file. Manipula does not automatically read records from the data file, but only when it is instructed to do so. The setting *CHECKRULES* takes care of checking the records after they have been changed. The checks carried out are defined in the rules section of the Blaise data model *Manure2*.

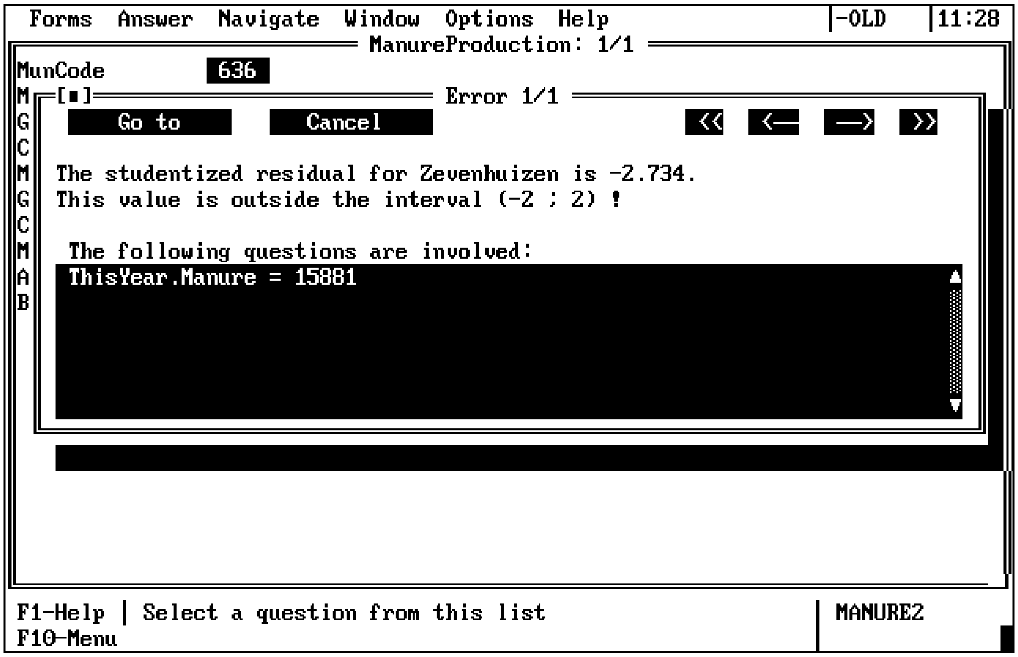
Figure 5.5. Manipula setup for computing regression quantities.

```
SETTINGS
    AUTOREAD = NO

 USES
    BlaiseData 'Manure2'

 UPDATEFILE
    UpFile: BlaiseData ('Manure2', BLAISE3)
 SETTINGS
    CHECKRULES = YES

 AUXFIELDS
    TYPE
       TReal = REAL[12, 3]

    BLOCK TMacroEdit
       FIELDS
          N    : 0..999
          Mean : TReal
          StDev: TReal
          A    : TReal
          B    : TReal
          RDev : TReal
    ENDBLOCK

 FIELDS
    MEdit: TMacroEdit
    Sx, Sy, Sxx, Syy, Sxy: REAL
    RecNum: INTEGER

 MANIPULATE
    FOR RecNum:= 1 TO UpFile.RECORDCOUNT DO
        UpFile.READNEXT
        Sy := Sy  + ThisYear.Manure
        Sx := Sx  + LastYear.Manure
        Sxx:= Sxx + SQR(LastYear.Manure)
        Syy:= Syy + SQR(ThisYear.Manure)
        Sxy:= Sxy + LastYear.Manure * ThisYear.Manure;
    ENDDO

    MEdit.N:= UpFile.RECORDCOUNT
    Sx:= Sx / MEdit.N
    Sy:= Sy / MEdit.N
    MEdit.B:= (Sxy - MEdit.N * Sx * Sy) / (Sxx - MEdit.N * SQR(Sx))
    MEdit.A:= Sy - MEdit.B * Sx
    MEdit.Mean:= Sx
    MEdit.StDev:= SQRT((Sxx - MEdit.N * SQR(Sx)) / MEdit.N)
```

The definition of the block *MEdit* in the setup is similar to the one in the Blaise data model. Again, the manipulate section contains three series of statements. The first series reads the Blaise data file, and computes some variables required for calculation of the regression quantities. The second series calculates the regression quantities, and assigns these values the corresponding variables in the block *MEdit*. The third series of statements resets the Blaise data file, and changes each record by reading it, copying the contents of the block *MEdit* to the block *MacroEdit*, and writing the record back to the file.

After the Manipula session has been successfully completed, all information is available to carry out a macro-editing operation. By setting the *ProcessMode* variable to the value *DataEditing*, and re-preparing the data model, the DEP can be run in data editing mode. All records with a studentized residual outside the interval (-2 ; 2) have an error attached to the field *Manure*. Figure 5.6 shows an example of an error message.

Figure 5.6. An example of a DEP error message



```
MunCode        636

The studentized residual for Zevenhuizen is -2.734.
This value is outside the interval (-2 ; 2) !

 The following questions are involved:
 ThisYear.Manure = 15881

F1-Help | Select a question from this list          | MANURE2
F10-Menu
```

The manure production of the Municipality of Zevenhuizen in the year 1990 is 15881. The studentized residual of this value is equal to -2.734. Since this value is less than -2, its flagged as an outlier.

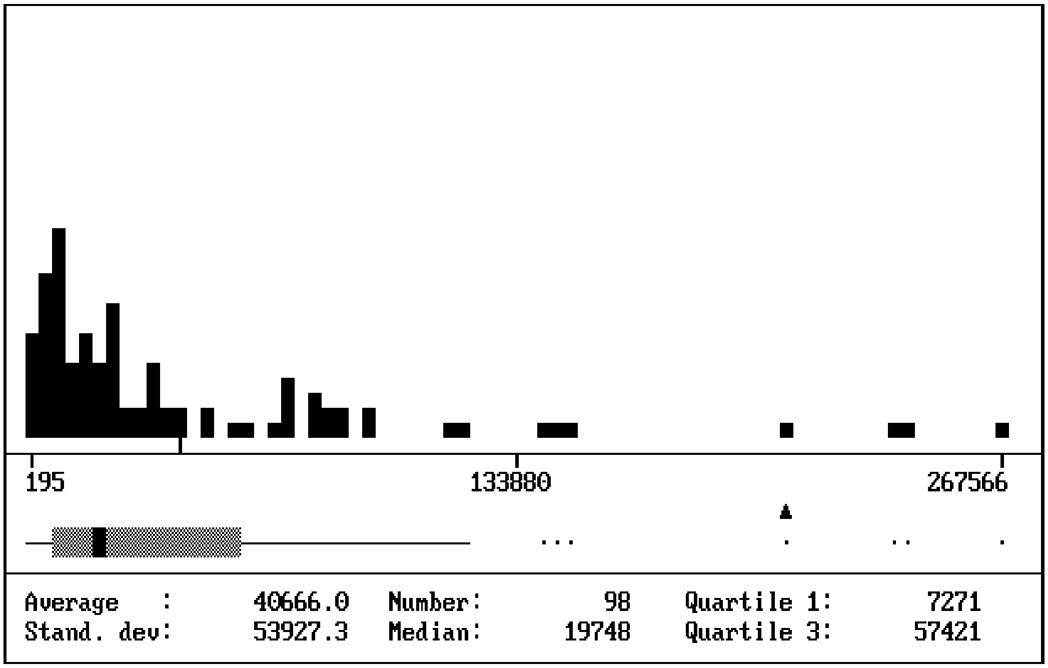
6. A front-end for macro-editing

The macro-editing approach in the previous sections has it limitations. One of the most important one is the lack of graphical facilities. In this section it is shown that it is not too difficult to build a macro-editing tool with more possibilities.

The basic idea is the following. Use the Data Selector of the Blaise Control Centre to select the variables that are to be included in the macro-edit analysis. Extract all values of the selected variables from the Blaise data file, and import these values in the macro-editing front-end tool. This tool displays all values graphically. The operator can selected a point in the graph. This causes the DEP to be activated for the corresponding record. Then the operator can change values in the fields of the record. After that, the point in the graph is adjusted accordingly.

The prototype of the macro-editing front-end can carry out both a one-variable and a two-variable analysis. First, the one-variable analysis is described. After having selected the analysis variable, a graphical overview appears on the screen containing all necessary information about the distribution. Figure 6.1 contains an example. The screen contains information about the manure production in 1990.

Figure 6.1. Analysis of the variable Manure in 1990



```
Average   :     40666.0  Number:        98   Quartile 1:      7271
Stand. dev:     53927.3  Median:     19748   Quartile 3:     57421
```

The top part of the screen contains the one-way scatterplot. It is the same scatterplot as displayed in figure 3.1. It is clear that the distribution of the manure production is very skew. There are many municipalities with a small manure production, and only a few municipalities with a large manure production. It might be better to carry out a transformation on this variable to transform its distribution into a normal one. However, this feature is not available in this prototype.

A box-and-whisker plot is displayed just below the scatterplot. It is a simplified version of the one in figure 3.2. The same scale is used. According to this plot, there are seven outliers. Part of these outliers may be due to the skewness of the distribution, but it may be worth while to look at the four extreme outliers.

Between the scatterplot and the box-and-whisker plot you can move a triangle-shaped cursor to the left and to the right (with the cursor-keys). By positioning the cursor under a point of interest, you can open the corresponding form. Since several forms may share the same location on the X-axis, first a list will appear containing field values identifying the forms. In this list, the form to be edited can be selected. The DEP will be activated for this form, and you change the value of the relevant variable. After the form has been closed, the screen with the scatterplot and the box-and-whisker plot re-appears. Note that as consequence of changing the value of the analysis variable, the square representing the form may have moved to a different location.

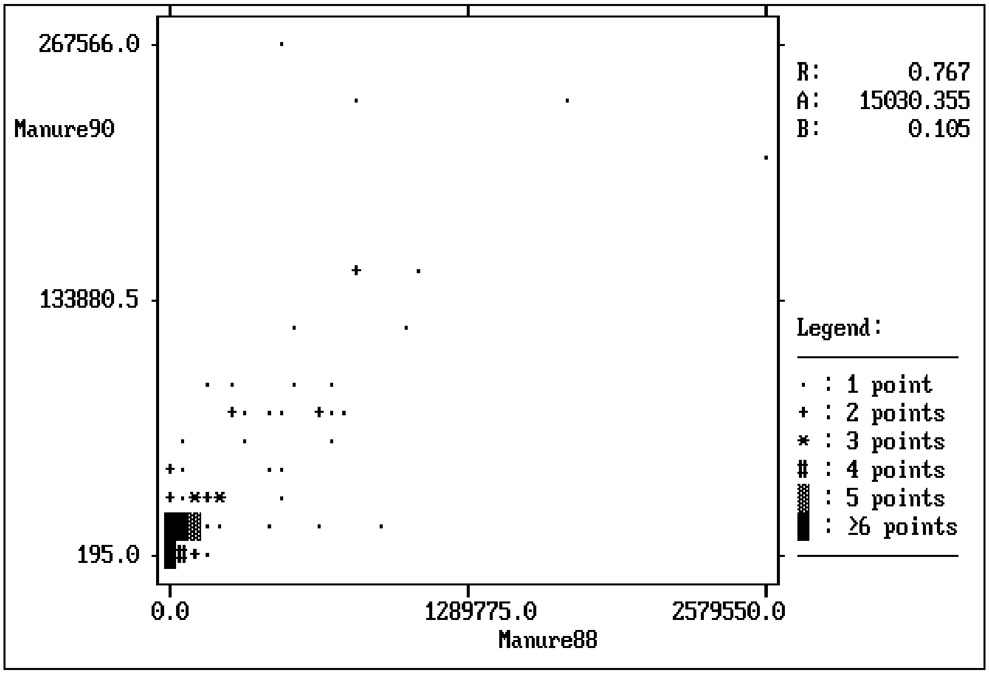
The aggregate information, like the box-and-whisker plot, and the numerical quantities at the bottom of the screen, are not effected by changes in individual values of the variables. They are only re-computed, if the program is instructed to do so. Note that re-computation of outliers after each individual change may lead to different results then waiting with re-computation after all original outliers have been taken care of.

For the two-way analysis, the prototype front-end produces a screen like the one in figure 6.2. It contains a two-way scatterplot of the variables involved. Note that overlapping points are indicated by special symbols (see the legend). The values of the correlation coefficient (R), and two regression coefficients (A and B) are displayed to the right of the plot.

A square cursor can be moved through the scatterplot (with the cursor keys). By positioning the cursor on a point of interest, you can open the corresponding form. Since several forms may share one location in the plot (due the same rounded X- and Y-co-ordinates), first a list will appear containing field values identifying the forms. In this list, the form to be edited can be selected. The DEP will be activated for this form, and you can change the value of the relevant variable(s). After the form has been closed, the screen with scatterplot and the box-and-whisker plot re-appears. Note that as consequence of changing the value of the variables involved, the dot representing the form may have moved to a different location.

Also here, the aggregate information (R, A and B) can be recomputed, but only if the program is instructed to do so. It is important to realise that re-computation of outliers after each individual change may lead to different results then re-computation after all original outliers has been taken care of.

Figure 6.2. Two-way analysis of the variable Manure in 1989 and 1990



7. Conclusion

Macro-editing can be interesting alternative at a time where resources to be spend on data editing must be minimised, at the same time maintaining a high level of data quality. The newest version of Blaise has some interesting possibilities for macro-editing, be it that they are somewhat limited due to the lack of graphical facilities. The paper also shows that a front-end can be implement with useful graphical macro-editing possibilities.

References

Chambers, J.M., Cleveland, W.S., Kleiner, B., and Tukey, P.A. (1983). *Graphical Methods for Data Analysis.* Duxburry Press, Boston, USA.

Granquist, L. (1990). *A Review of some Macro-editing Methods for Rationalizing the Editing Process.* Proceedings of the Statistics Canada Symposium 90, pp. 225-234.

Tukey, J.W. (1977). *Exploratory Data Analysis.*