

CONFERENCE OF EUROPEAN STATISTICIANS

UN/ECE Work Session on Statistical Data Editing

(Rome, Italy, 2-4 June 1999)

Topic (ii): Generalized software packages for statistical data editing, their evaluation

FROM CHERRYPI TO SLICE

Submitted by Statistics Netherlands¹

Contributed paper

I. INTRODUCTION

1. In the last few years the Department of Statistical Methods at Statistics Netherlands has developed two prototype computer programmes for statistical data editing. One of these computer programmes, CherryPi, was designed to edit and impute numerical data automatically. The other programme, MacroView, is a tool to detect outliers in a data set in a graphical manner. After an outlier has been detected, the corresponding data can be adjusted interactively.

2. Recently, the Department of Statistical Informatics at Statistics Netherlands has started with the development of a general software package, called SLICE (Statistical Localisation, Imputation and Correction of Errors), for edit and imputation. SLICE will contain several edit and imputation modules. Examples of such modules are improved versions of CherryPi and MacroView. SLICE will conform to modern software standards. For instance, the architecture SLICE is such that it will be easy to add additional modules. SLICE itself is planned to become a module of Blaise, the integrated survey processing system developed by Statistics Netherlands. More information about Blaise can be found in the *Blaise Reference Manual* (1998) and the *Blaise Developer's Guide* (1998).

3. At the moment of writing this paper in early 1999, work has been focused on re-programming and improving CherryPi. CherryPi is the first, and so far the only module in SLICE. Section II of this paper briefly describes the methodology of CherryPi. The CherryPi-module in SLICE is examined in Section III. Section IV concludes the paper with a description of a number of modules that will be added to SLICE in (near) future.

4. Pierzchala (1995) gives a good overview of software for computer-assisted editing, graphical editing and automatic editing. For the views of Statistics Netherlands on combining several editing techniques we refer to De Jong (1996) and Van de Pol et al. (1997). We refer to Van de Pol and Bethlehem (1997) for more information on the views of Statistics Netherlands on the future of data editing in general.

II. CHERRYPI

5. The aim of statistical data editing is to modify incorrect records, i.e. data from individual respondents, in such a way that statistical results based on the edited data are acceptably accurate. It is

¹ Prepared by Ton de Waal and Hans Wings.

important to note that not all errors have to be removed from the original data set. This observation allows one to automate part of the edit and imputation process.

6. To edit and impute numerical data automatically, CherryPi basically performs three steps. Firstly, CherryPi identifies the erroneous records. Secondly, CherryPi identifies the faulty values in each erroneous record. Thirdly, CherryPi replaces the faulty values by better values. These three steps are discussed in Sections II.1 to II.3.

7. After CherryPi has finished its work, an edited data set is returned. This edited data set contains only records that satisfy the edit checks. CherryPi also returns some information on the modifications that have been carried out per record as well as per variable.

II.1 Identifying erroneous records

8. To identify erroneous records, edit checks have to be specified by subject-matter specialists. If an edit check is failed, the record is considered incorrect; if all edit checks are satisfied, the record is considered correct. CherryPi modifies all records that are considered incorrect.

II.2 Identifying faulty values in a record

9. After an incorrect record has been found, the next step is to identify its faulty fields. To identify faulty values in an erroneous record automatically some guiding principle is needed. In an important paper on automatic edit and imputation Fellegi and Holt (1976) propose such a guiding principle. According to the Fellegi-Holt paradigm the data of a record should be made to satisfy all edit checks by changing the values of the fewest possible number of variables. Missing values are always imputed.

10. CherryPi is based on a generalised version of the Fellegi-Holt paradigm: the data of a record is made to satisfy all edit checks by changing the values of the variables with the smallest possible sum of *confidence weights*. A confidence weight of a variable is a positive number that expresses the confidence one has in the correctness of the values of this variable. A high confidence weight corresponds to a variable of which the values are considered trustworthy, a low confidence weight to a variable of which the values are considered not so trustworthy. If all confidence weights are equal to the same positive number, say 1, the original Fellegi-Holt paradigm is obtained. Again, missing values are always imputed.

11. In some cases application of the (generalised) Fellegi-Holt paradigm yields several optimal ways to modify the data. In CherryPi an additional criterion can be used to select one of these optimal ways.

12. The (generalised) Fellegi-Holt paradigm seems to be a very natural guiding principle to identify the faulty fields. Unfortunately, application of the paradigm turns out to be quite complicated. In fact, it is so complicated that most algorithms and computer programmes for automatic edit and imputation are suitable for either categorical data or numerical data, but not for both simultaneously. When developing CherryPi we decided to develop an algorithm and corresponding software for numerical data only.

13. Some pre-processing is done in CherryPi before the faulty values are actually identified by means of the above-mentioned algorithm. During this pre-processing phase potential transcription and typing errors are identified. For more details about this phase of CherryPi we refer to Van de Pol, Bakker and De Waal (1997).

II.3 Imputing

14. After the faulty values in a record have been identified, the corresponding variables are imputed. CherryPi takes care of imputation in two steps. In the first step regression imputation is used to impute the faulty fields. The edit checks are not taken into account during this first step. Only statistical considerations determine which values are imputed.

15. After the preferred imputation has been carried out, the resulting record may still fail some edit checks. Therefore, in a second imputation step the imputed values are adapted slightly in such a way that the resulting record satisfies all edit checks. A practical problem is that changes in different kinds of variables may have to be compared to each other. For example, the variable ‘Number of employees’ and the variable ‘Turnover’ are measured in different units, and therefore changes in these variables cannot be compared directly. To facilitate this comparison the subject-matter specialist has to assign a numerical value, called COST (see Section III.4), to each variable. In the example above, one may, for instance, assign the average turnover per employee to the COST-attribute of ‘Number of employees’ to compare changes in ‘Number of employees’ to changes in ‘Turnover’. For more details about imputation in CherryPi we refer to De Waal (1996).

III. THE CHERRYPI-MODULE IN SLICE

16. The CherryPi-module in SLICE is based on the same methodology as the original version of CherryPi, but, for instance, the user interface of SLICE is much better than the old one of CherryPi. In CherryPi, edit checks had to be specified via a complicated graphical user interface; in SLICE edit checks can be specified via a language similar to the Blaise language.

17. Another improvement is the imputation module. The imputation module of SLICE is an extension of the imputation module of CherryPi. In CherryPi only one predictor was allowed per variable, in SLICE an arbitrary number of predictors is allowed. Moreover, in CherryPi each imputation rule holds for all records, in SLICE conditional imputation rules can be defined. These conditional imputation rules are only applied conditional on the values of certain variables.

18. The syntax of a SLICE project file is largely based on the Blaise language. Some of the features of SLICE require some new keywords, however. A SLICE project file consists of 6 different sections. These sections start with the keywords **SLICE**, **USES**, **SETTINGS**, **FIELDSELECTION**, **RULES**, and **IMPUTE**. With the exception of the section starting with the keyword **SETTINGS** all sections are required.

19. We illustrate a SLICE project file by means of the example below. This example is taken from Hölgens and Wings (1998).

```

SLICE
  ISSCorrection "Automatic correction of ISS: Income & Savings Survey"

USES
  IncomeAndSavings 'j:\surveys\metadefinitions\iss'

SETTINGS
  LOGFILE = 'd:\temp.log'
  LOW = 0.95

FIELDSELECTION
  HHSize
  Person[1].Gender      (ALIAS = Gender1  TRUST = CERTAIN)
  Person[2].Gender      ALIAS = Gender2
  Person[1].Age         (ALIAS = Age1     TRUST = HIGH)
  Person[2].Age         (ALIAS = Age2     TRUST = 2.25)
  Person[1].MarStat     (ALIAS = MarStat1 TRUST = CERTAIN)
  Person[1].Income      (ALIAS = Income1  TRUST = LOW    COST = 1.5)
  Person[1].Savings     (ALIAS = Savings1 TRUST = MEDIUM COST = 2.0)

RULES
  HHSize > 1
  IF Gender1 = Man THEN
    IF MarStat1 = Yes THEN
      ORD(Gender2) = 2
      Income > 50000
      Savings1 > 0.05*Income
      Age2 IN [30 .. 45]
    ENDIF
  ENDIF

```

```

ENDIF

```

```

IMPUTE
  Income := 0.45 * Savings1 + 12000
  IF MarStat1 = Yes THEN
    IF Gender1 = Man THEN
      ORD(Gender2) = 2
    ELSE
      ORD(Gender2) = 1
    ENDIF
  ENDIF

```

III.1 Defining the project: SLICE

20. The keyword **SLICE** followed by the name of the project and optionally also by a short description of the project indicates that the file should be interpreted as a SLICE project file. In our example, the name of the project is 'ISSCorrection', and a short description is 'Automatic correction of ISS: Income & Savings Survey'.

III.2 Specifying the data model: USES

21. The data model of the data that are to be edited must be specified in the section starting with the keyword **USES**. The keyword **USES** must be followed by the name of the data model. This data model should be specified in Blaise meta-data format. Optionally, the name of the data model may be followed by the corresponding file name. If a file name is specified SLICE uses the file *file name.~mi*, if no file name is specified SLICE uses the file *project name.~mi*. In our example the data model is called 'IncomeAndSavings'. The file containing this data model is called 'j:\surveys\metadefinitions\iss.~mi'.

III.3 Specifying the project settings: **SETTINGS**

22. Optionally, project settings may be specified. To change the default settings one should use the keyword **SETTINGS**. One can change the setting of five parameters.
23. By default, no log file is created. If one wants to create a log file, the keyword **LOGFILE** followed by the name of the log file should be used. By default, no pre-processing is done to detect transcription and typing errors. If one wants to pre-process the data, **CHECKTYPOS** should be given the value **YES**.
24. To detect faulty values in an incorrect record **SLICE** modifies this record in such a way that all edit checks are satisfied by changing the values of the variables with the smallest possible sum of confidence weights. In **SLICE** three default confidence weights are defined: **HIGH**, **MEDIUM** and **LOW**. **HIGH** (default value 2) may be used for variables with a high confidence weight, **MEDIUM** (default value 1) for variables with a ‘normal’ confidence weight, and **LOW** (default value 0.5) for variables with a low confidence weight. The values of these three parameters may be changed in the **SETTINGS** section. In our example a log file must be created, and the constant **LOW** is set to 0.95.

III.4 Specifying additional information for selected fields: **FIELDSELECTION**

25. **SLICE** needs more meta-information than the Blaise data model can provide. The additional meta-information should be specified after the keyword **FIELDSELECTION**. After **FIELDSELECTION** all variables that are used in the edit and imputation process should be mentioned.
26. If one wants to use an alias for a variable name one should use the keyword **ALIAS** followed by the alias itself. To specify the confidence weight of a variable one should use the keyword **TRUST** followed by the value of the confidence weight. **TRUST** may be given the values **CERTAIN**, **LOW**, **MEDIUM**, **HIGH**, or a numerical value. If the value **CERTAIN** is given to **TRUST** the values of the corresponding variable may not be modified. **LOW**, **MEDIUM** and **HIGH** are constants that have already been described in the previous subsection. If no value is assigned to **TRUST** the default value **MEDIUM** is used.
27. The value of the attribute **COST** may be specified for each field. This value is used when the record obtained after the first imputation step does not satisfy all edit checks, and has to be modified in order to satisfy these edit checks (see Section 2.3). The attribute **COST** can be used to compare different variables. If no value is assigned to **COST**, the default value 1 is used.
28. For instance, in our example the field ‘Person[1].MarStat’ is given the alias **MarStat1**. This field may not be changed. The field ‘Person[1].Income’ is given the alias **Income1**. This field is not considered very trustworthy, hence **TRUST** is given the value **LOW**. To compare changes in the value of this field to changes in values of other fields, **COST** is given the value 1.5.

III.5 Specifying the edit checks: **RULES**

29. After the keyword **RULES** the edit checks can be specified. This can be done in the same way as in Blaise. Edit checks that should hold for all records as well as edit checks that should hold conditional on the values of certain variables can be specified.
30. **SLICE** cannot handle all edit checks that can be specified in Blaise. From all edit checks that are specified, **SLICE** automatically selects the edit checks it can deal with. Warnings are given for those edit checks that cannot be handled by **SLICE**.
31. In our example the value of ‘**HHSize**’ should be larger than 1 in all records. For those records for which the value of ‘**Gender1**’ equals ‘**Man**’ and the value of ‘**MarStat1**’ equals ‘**Yes**’, ‘**ORD(Gender2)**’ should be 2, the value of ‘**Income**’ should be larger than 50,000, the value of ‘**Saving1**’ should be larger than 0.05 times the value of ‘**Income**’, and the value of ‘**Age2**’ should lie between 30 and 45 (including 30 and 45).

III.6 Specifying the imputation model: IMPUTE

32. The imputation model should be specified after the keyword **IMPUTE**. For each variable an imputation rule may be defined. One can specify imputation rules that hold for all records as well as imputation rules that are conditional on the values of certain variables.

33. In our example 0.45 times the value of 'Savings1' plus 12,000 is imputed for the value of 'Income'. If the value of 'MarStat1' is 'Yes' and the value of 'Gender1' is 'Man' then the value 2 is imputed for 'ORD(Gender2)', if the value of 'MarStat1' is 'Yes' and the value of 'Gender1' is not 'Man' then the value 1 is imputed for 'ORD(Gender2)'. Of course, the fields 'Income' and 'ORD(Gender2)' are only imputed in those records for which the original values have been identified as faulty or are missing.

IV. FUTURE MODULES OF SLICE

34. Several new modules of SLICE are planned to be developed. In this section we briefly discuss three of these modules.

IV.1 MacroView

35. MacroView is a computer programme for graphical macro-editing. It requires a data set that has to be edited and, preferably, a similar data set of a previous period. It also requires meta-information on the variables in these data sets. To calculate publication figures the raising weights of the records in the data set(s) are needed. Finally, one should describe the so-called stratification of the publication figures, i.e. one should describe what kind of figures are published. The meta-information can be specified by indicating for each variable whether it is the record identification number, a stratification variable, a raising weight, or a normal variable.

36. MacroView consists of three levels: the macro-level, the meso-level and the micro-level. At the macro-level implausible publication figures are detected, subsequently at the meso-level implausible records and values are detected, and finally at the micro-level the implausible values can be corrected. We briefly discuss the three levels in turn. For more information on MacroView we refer to De Waal (1998b).

IV.1.1 The macro-level

37. At the macro-level the publication figures of this period are compared to those of the previous period. The user must specify a threshold percentage p . A publication figure in this period that differs more than the specified percentage from its value in the previous period is considered suspicious, else it is not considered suspicious. Colours are used to indicate the status of a publication figure in this period in comparison to its value in the previous period. Green indicates that the present value of the publication figure differs less than $p\%$ from the value in the previous period, red that the present value is more than $p\%$ larger than the value in the previous period, and purple that the present value is more than $p\%$ smaller than the value in the previous period. Yellow indicates that the sign of the publication figure in this period has changed compared to its value in the previous period. Finally, blue indicates that there were no data contributing to this publication stratum in the previous period.

IV.1.2 The meso-level

38. After one has decided to edit (part of) the records contributing to a certain publication figure, one enters the meso-level. At the meso-level multivariate outliers can be detected. These outliers often correspond to implausible records, containing implausible values. Usually, one assumes that non-outlying records do not contain serious errors, i.e. errors that have a significant influence on the published figures. These non-outlying records are therefore often not examined in detail.

39. MacroView enables the user to detect outliers in two different ways. Firstly, the user can select a number of variables, and then make scatterplots of these variables. The outliers can subsequently be detected visually. Secondly, MacroView supports a mathematical algorithm to detect multivariate outliers automatically.

IV.1.3 The micro-level

40. After one has detected outlying records and fields one can enter the micro-level of MacroView. At the micro-level the data of a particular record that seems implausible can be examined. If some values of this record are considered to be incorrect, they can be modified. All corrections that have been made are taken track of in a log file. The impact of corrections made at the micro-level can immediately be seen at the meso-level.

IV.2 Localisation of faulty values in a mix of numerical and categorical data

41. The present error localisation module of SLICE can only handle numerical data. Many data sets however consist of a mix of numerical and categorical data. We therefore would like to be able to deal with such a mix of numerical and categorical data. Although our focus will remain on automatic edit and imputation of numerical data, we simultaneously would like to allow a limited number of categorical variables. Much research has been devoted to this subject at Statistics Netherlands in the last few years (cf. De Waal, 1997a, 1997b and 1998a). The developed algorithms are however very complex – and hence difficult to implement – and are likely to be (too) time-consuming in practice. At the moment alternative algorithms are being studied.

IV.3 Hot-deck imputation

42. Hot-deck imputation, where faulty values in an incorrect record are replaced by values in a similar but correct record, is often considered to be better than regression imputation. Hence a module for a hot-deck method would be highly desirable. Fortunately, Statistics Netherlands is participating in a European project on imputation. One of the aims of this project is to develop a hot-deck imputation method based on Automatic Interaction Detection (AID) (cf. Sonquist, Baker and Morgan, 1971). The basic AID methodology will be extended with a number of new developments in order to make AID better suited as a tool for hot-deck imputation. The developed algorithm will be implemented in a stand-alone computer programme. A slightly adapted version of this computer programme is planned to become a module of SLICE.

References

- Blaise Reference Manual*, 1998, Department of Statistical Informatics, Statistics Netherlands, Heerlen.
Blaise Developer's Guide, 1998, Department of Statistical Informatics, Statistics Netherlands, Heerlen.
 De Jong, W.A.M., 1996, Designing a Complete Edit Strategy; Combining Techniques. Report, Statistics Netherlands, Voorburg.
 De Waal, T., 1996, CherryPi: A Computer Program for Automatic Edit and Imputation. Report, Statistics Netherlands, Voorburg.
 De Waal, T., 1997a, Cutting Plane Algorithms for Optimal Automatic Error Localization. Report, Statistics Netherlands, Voorburg.
 De Waal, T., 1997b, The Error Localisation Problem as a Dynamic Disjunctive-Facet Problem. Report, Statistics Netherlands, Voorburg.
 De Waal, T., 1998a, Mathematical Programming Techniques for Solving the General Error Localisation Problem. Report, Statistics Netherlands, Voorburg.
 De Waal, T., 1998b, An Introduction to CherryPi and MacroView. Report, Statistics Netherlands, Voorburg.
 Fellegi, I.P. and D. Holt, 1976, A Systematic Approach to Automatic Edit and Imputation. *Journal of the American Statistical Association*, **71**, 17-35.
 Hölsgens, H. and H. Wings, 1998, SLICE Projectfile. Internal memo, Statistics Netherlands, Heerlen.
 Pierzchala, M., 1995, Editing Systems and Software. In: *Business Survey Methods* (Ed. Cox, Binder, Chinnappa, Christianson, Colledge and Kott), John Wiley & Sons, Inc.

- Sonquist, J.N., E.L. Baker and J.A. Morgan, 1971, Searching for Structure. Institute for Social Research, University of Michigan.
- Van de Pol, F., F. Bakker and T. De Waal, 1997, On Principles for Automatic Editing of Numerical Data with Equality Checks. Report, Statistics Netherlands, Voorburg.
- Van de Pol, F. and J. Bethlehem, 1997, Data Editing Perspectives. *Statistical Journal of the United Nations ECE*, **14**, 153-171.
- Van de Pol, F., A. Buijs, G. Van der Horst and T. De Waal, 1997, Integrating Automatic Data Editing, Computer-Assisted Editing and Graphical Macro-Editing. Report, Statistics Netherlands, Voorburg.