

**UNITED NATIONS STATISTICAL COMMISSION and
ECONOMIC COMMISSION FOR EUROPE**

CONFERENCE OF EUROPEAN STATISTICIANS

Work Session on Statistical Data Editing
(Vienna, Austria, 21-23 April 2008)

Topic (i): Editing of data acquired through electronic data collection

**SELECTIVE AUTOMATIC EDITING OF MIXED MODE QUESTIONNAIRES FOR
STRUCTURAL BUSINESS STATISTICS**

Invited Paper

Prepared by Jeffrey Hoogland and Roos Smit, Statistics Netherlands

Abstract: Statistics Netherlands uses electronic questionnaires for structural business statistics (SBS) from statistical year 2006. Part of the enterprises still returns filled-in paper questionnaires. Furthermore, SBS questionnaires for 2006 were redesigned to decrease the response burden. These changes induced us to improve selective and automatic editing procedures. We changed the treatment of missing values and implemented the latest version of the automatic editing program SLICE. In this paper we state the implications of these changes, examine the performance of SLICE, and investigate mixed mode effects.

I. INTRODUCTION

1. Statistics Netherlands implemented electronic questionnaires for SBS 2006. Non-responding enterprises are given the possibility to fill in a redesigned paper questionnaire. The lay-out of the old paper questionnaire caused measurement errors and a large respondent burden (Giesen & Hak, 2005). An advantage of electronic observation is that data have already been keyed in by respondents and subtotals can be generated automatically. This should lead to fewer edit failures to be solved in the editing process. Furthermore, SBS questionnaires were redesigned to decrease the response burden. They contain smaller sections and fewer variables for medium-sized and large enterprises.

2. Since SBS 2000, advanced methods for selective and automatic editing are used (De Jong, 2002; Hoogland, 2006). Plausibility indicators are used to detect records with potential influential errors. These records are edited interactively, while the remaining records are edited automatically. Changes in questionnaires have many consequences for editing procedures. In this paper we discuss some consequences and choices we made. We also implemented improved methodology and software.

3. In Section 2, we discuss the changes in the questionnaires for SBS 2006. Section 3 deals with consequences for plausibility indicators. In Section 4, we explain consequences for automatic editing procedures and we give some details regarding the performance of SLICE 1.6. In Section 5, we investigate mixed mode effects. Finally, some conclusions are drawn in Section 6.

II. ELECTRONIC QUESTIONNAIRES

4. For SBS 2006, we first developed questionnaires with a top-down structure, starting with the profit-and-loss account. In this design, respondents first have to fill in the profit-and-loss account. They are allowed to use their own administrative definitions. Then a respondent is asked to indicate the relation between their administrative definitions and our definitions used in the remainder of the questionnaire. The subsequent blocks of questions relate to employed persons, business profit, and business costs. Top-down questionnaires were tested in agreement with a small number of respondents. The required time for respondents to fill in a questionnaire dropped significantly, because they did not have to make the profit-and-loss account consistent with the rest of the questionnaire (Giesen & Hak, 2005). However, because of large implications for our editing process, we decided to continue with bottom-up questionnaires. For each block we start with detailed questions, which have to add up to the total for that block. We end with the profit-and-loss account, which has to be consistent with other blocks.

5. While maintaining a bottom-up approach, the number of questions in an SBS questionnaire was decreased substantially. For large and medium-sized enterprises, we frequently asked each item of a question separately. For instance, regarding the costs of machinery, we used to ask three parts. Namely, rent/lease of machinery, maintenance/repair of machinery, and other machinery-related costs. These subdivisions of questions could be considered annoying by respondents and therefore are no longer asked separately. However, one aim was to prevent the respondent from forgetting parts of the cost for machinery. Furthermore, it gave an extra edit check, namely whether the three parts sum up to the total cost for machinery.

6. All enterprises sampled for SBS 2006 received a letter with internet access codes for the electronic questionnaire. Enterprises which encountered technical problems or had no internet access could receive a paper questionnaire on request. Enterprises which did not respond until the last reminder were also offered the opportunity to fill in a paper questionnaire. We therefore have a mixed mode questionnaire. When filling in the electronic questionnaire, subtotals and questions recurring in the profit-and-loss account are filled in automatically.

III. PLAUSIBILITY INDICATORS

7. We edit records using the principles of selective editing, cf. Granquist & Kovar (1997). Records are therefore grouped into relatively homogeneous subgroups. These *edit cells* are an intersection of a publication cell and size class. There are three size classes: small (<20 employed persons), medium (20-100 employed persons), and large (>100 employed persons). For each edit cell, we calibrate plausibility indicators for year t using year $t-1$ and $t-2$ (Hoogland, 2006). We can then determine for each record for year t whether it is plausible enough for automatic editing. Large enterprises are always edited manually. However, plausibility indicators are still used as a tool for editors to detect influential errors.

8. To assess plausibility, we estimate the correct values of variables in a record. These estimations are called *reference values* and are mostly based on year $t-1$. A consequence of a new questionnaire is that some variables are not comparable between year t and year $t-1$. Variables of 2004-2005 have to be transformed to the 2006 format to obtain reference values and calibrate plausibility indicators. However, a one-to-one transformation does not exist for some variables. Raw and edited data used to compute and calibrate plausibility indicators may therefore not be available. However, the unavailable variables are often not crucial in determining plausibility. Nonetheless, the result is a smaller subset of usable variables and some influential errors might go undetected. For some important variables, we approximated reference values using related variables for year $t-1$. For instance, gross earnings no longer include received sickpay for SBS 2006. We therefore added the received compensation for 2005 to the gross earnings for 2005, because sickpay makes up about 95% of compensation.

IV. AUTOMATIC EDITING

A. SLICE 1.6

9. When a record for SBS 2006 is considered plausible, it is edited with the SLICE 1.6 program developed at Statistics Netherlands. It contains the module CherryPie (Quere, 2000; De Waal & Quere, 2003). This module localises errors in a record on the basis of a number of edit rules that show the mathematical relationships between variables. It also contains an imputation module, which can apply mean, ratio, or regression imputation in the case of erroneous values. Manually edited data is used to estimate the imputation model. Finally, it contains a module that corrects imputed values that violate edit rules or imputes remaining missing values that violate edits (De Waal, 2000).

10. SLICE identifies erroneous values of one or more variables when edit rules are violated. Usually there are several possibilities for editing a record. This requires a choice, which is made according to the *principle of Fellegi and Holt*, see Fellegi & Holt (1976). The principle states that the best solution for editing a record is to change as few variables as possible.

11. In practice, some variables contain fewer errors than others, or we do not want to change them as frequently as other variables. Each variable in CherryPie is therefore weighted for reliability. This implies that a change in one variable can weight a number of times heavier than a change in another variable. The best solution for the record is obtained by minimising the sum of the reliability weights of the variables changed, in such a manner that all edit rules are satisfied. The resulting constrained minimisation problem is solved with a branch-and-bound algorithm (De Waal, 2002; De Waal and Quere, 2003).

12. To solve the error localisation problem, this algorithm passes through a binary tree for each record. In each node, a variable is chosen based on heuristics. There are two options:

- 1) A value of a variable is correct. This variable is fixed and the set of edits is adjusted.
- 2) A value of a variable is erroneous. This variable is eliminated from the set of edits.

This means that the set of edits is updated for each considered node. A solution is found, if the remaining edits for a node are true by definition.

13. In some cases, the amount of edits at a node grows quickly and the performance of SLICE deteriorates. We therefore set a maximum for the number of edits at a node. We also set a maximum for the number of erroneous values, the number of missing values and the computation time for a record. Whenever SLICE ends up in a branch of the binary tree where the maximum number of edits or erroneous values is exceeded, this branch is cut off. The algorithm also cuts off a branch in several other cases (De Waal and Quere, 2003) and can be quite fast.

B. Modifications of automatic editing for SBS 2006

14. Because of changes in the questionnaires and some problems regarding SLICE, a number of improvements were implemented for SBS 2006. Since a large number of variables and/or variable names changed, the old edit rules no longer sufficed. In the past, the edit rules used for SLICE had a different form than edit rules used for interactive editing. This originated from the inability of SLICE 1 to interpret 'IF statement 1 THEN statement 2' edit rules when the value of a variable in statement 1 is erroneous. In 2004 SLICE 1 was updated to SLICE 1.5, which has the ability to interpret many types of edit rules for numerical and categorical variables (De Waal and Quere, 2003). General (BLAISE) edit rules were already used for interactive editing. Nonetheless, due to performance problems and lack of time, they were not used for SLICE. In an effort to save time and simplify the administration of edit rules for SBS 2006, we decided to use BLAISE edit rules for SLICE. These edits were already translated to the 2006 format for interactive editing purposes and had been subjected to extensive testing.

15. To obtain acceptable SLICE solutions for SBS 2006, some 2005 edit rules formulated specifically for SLICE need to be added. These mainly consist of ratio rules, i.e. rules which specify a range for a certain ratio of two variables. Ratio rules are used to compel SLICE to obey certain proportions. For example, in the rental business, the main activity is rental without operating personnel. Turnover from this activity is supposed to make up at least 80% of total turnover. Since exceptions very rarely do occur, this is a soft edit rule for interactive editing as otherwise exceptions would no longer be possible. For SLICE, on the other hand, this is a hard edit rule, because an exception is far less likely than a mistake. On the whole, the use of this ratio rule for SLICE will cause few unjust corrections and a majority of right corrections.

16. A second change is the way we deal with entries that are left empty by the respondent. Since SBS 2000, all empty entries were automatically filled with zeros before running SLICE. For SBS 2006, we no longer fill these entries with zeros in most cases. A filled-in zero was probably meant as a zero, while an empty entry might be an erroneously missing value. Furthermore, SLICE considers zeros and missing values as two different things. When editing a record, SLICE will mark all of the missing values as erroneous first, which may result in solving all edit failures. SLICE will otherwise try to solve edit failures by marking filled-in entries as erroneous. After SLICE has a complete set of erroneous variables, it will impute these variables. If necessary, these imputations are adapted to obey edit rules. In this approach all missing values are imputed, which results in more realistic solutions as is shown in table 1. In this example we consider the addition of extra costs and leave other rules aside. It shows how leaving entries empty can drastically change the way SLICE chooses its solution. The solution with missing values as input results in less extreme values and a more likely solution.

Table 1. Imputation of missing values with SLICE 1.6.

Variables	Observed values	Solution if missing values are first set at zero	Solution if missing values are edited by SLICE
Housing costs	-	100	60
Gas, water costs, etc.	-	0	20
Transportation costs	-	0	10
Communication costs	-	0	8
Costs for machinery	-	0	1
Advertising costs	-	0	1
Total extra costs	100	100	100

17. For some variables, a missing value can often be considered as a zero value. In this case, it is less desirable that SLICE always imputes missing values. We therefore use different states of protection. They determine whether variables are allowed to be changed and whether missing values are set at zero before entering the SLICE module. The protection status values are:

- 1) SLICE is not allowed to change filled-in values. Missing values are set to zero.
- 2) SLICE is only allowed to change missing values or values equal to zero.
- 3) Missing values are set to zero.
- 4) SLICE is only allowed to change values equal to zero. Missing values are set to zero.

18. The protection status values are used for different situations. Status value 1 is used to protect variables that usually are filled in very well by respondents, and except for being in an addition do not depend on other variables (for instance depreciations). Without the status value, these variables would often be changed by SLICE, because of their independence and therefore lack of rules. This lack of rules means that SLICE can easily change these variables and still obey the edit rules. Status value 2 is similar to status 1, the difference being that status 2 only protects filled in values greater than zero. This is useful for variables which depend on other variables. For instance, cost of temporary employees depends on the number of temporary employees. These variables are therefore linked in an edit rule. If left without a protection status value, SLICE might remove the cost of temporary employees to prevent having to adjust the number of temporary employees. Status value 3 is designed for variables that are overimputed. This happens to variables that are often left empty when zero is meant, for example, variables that have their own page in the questionnaire or variables that are very rarely reported in a branch. For provisions, both

of these conditions apply. Since provisions are very rare and have their own page, they are often skipped by respondents. In this case, empty usually does mean zero. Status value 4 is a combination of status value 3 and status value 2. This means that it only protects values unequal to zero and fills an empty entry with zero. This is useful for variables which tend to be overimputed, while non-zero observed values are often correct.

19. After analysing SLICE solutions, it became clear that SLICE prefers imputing some less often reported variables with small reliability weights. This created a large bias for these variables. We therefore change two types of parameters for SLICE. First, the reliability weights are changed. A variable representing the total of an addition is still considered the most reliable. However, variables within an addition are handled differently, as is shown in table 2. Variables which generally have small values, like the “other” variables, now obtain large weights. On the other hand, variables which normally have large values now obtain small weights. This is done to obtain a smaller bias. If a change needs to be made, the change is relatively small on a large value. Second, the way of prioritising between two variables with equal weights is changed. SLICE used to solve this problem in a reproducible way; it always chose the first solution. To obtain more realistic proportions between variables, SLICE now randomly chooses between variables with equal weights. Suppose that values are not missing or considered as zeros. SLICE then tries to satisfy an addition edit rule by changing a value that is generally the largest one within the addition. If there are more solutions with the same weight, the value to be changed is randomly chosen.

Table 2. Influence of reliability weights on SLICE solution for car trade.

Variables	Old weights	New weights	Values	Old solution	New solution
Turnover from service	3	1	0	0	0 or 100
Turnover from retail	2	1	0	0	0 or 100
Turnover from wholesale	1	2	0	100	0
Other proceeds	1	3	0	0	0
Total proceeds	4	4	100	100	100

20. Using most of the edit rules for interactive editing for SLICE has several consequences. For a specific trade, the number of edit rules is about doubled and some edit rules are quite complex. Because of improvements in edit rules, SLICE is forced to provide a solution that resembles the way an editor would solve the record. The use of protection status values makes SLICE better controllable. Furthermore, the adapted reliability weights and new approach for missing values make influential errors introduced by SLICE less likely. All of this results in more realistic solutions by SLICE. However, the question is whether SLICE can solve the same amount of records. This is discussed in paragraph IV.D.

C. Testing of SLICE

21. Before implementing SLICE 1.6, it was tested extensively (see table 3). The testing consisted of several waves, starting with test runs on a PC using BLAISE-databases. The acceptance environment is similar to the production environment. It consists of a NT network, an SQL-database, BLAISE Meta Information files, and Visual basic code that controls SLICE and the SQL-database. When the performance of SLICE is acceptable, the final settings from the acceptance environment are copied to the production environment.

22. In test wave 2, severe performance problems were noted for SBS 2003 records using BLAISE-edits. A new version of SLICE was developed by the Methodology department, but the managers of business statistics had lost their interest. This gave a large delay in the use of BLAISE-edits for automatic editing of structural business statistics. A few years later, SLICE 1.6 was tested with BLAISE-edits to see whether the performance problems noted earlier reoccurred. The new reliability weights were introduced in test wave 5 and the protection status values were introduced in test wave 6.

Table 3. Test waves for SLICE 1.5 and 1.6.

Wave	SLICE version	Year	System	Data
1	1.5	2004	PC	All kinds of datasets
2	1.5	2004	Acceptation environment	SBS 2003 records
3	1.6	2004	PC	All kinds of datasets
4	1.6	2007	PC	SBS 2005 records for car trade transformed to 2006 format
5	1.6	2007	PC	SBS 2006 records for car trade
6	1.6	2008	Acceptation environment	Large selection of SBS 2006 records
7	1.6	2008	Production environment	Eventually all SBS 2006 records meant for SLICE

D. Performance of SLICE 1.6

23. For the fifth test wave, we consider 203 medium-sized enterprises for SBS 2006 Car trade. This data set contains 85 variables, mainly integers. We use 140 edit rules, including 13 'IF statement 1 THEN statement 2' edits and 18 ratio edits. We test with data containing missing values and with data where all missing values are set at zero. We show results for the following parameters.

Maximum of:

- missing variables: 100
- erroneous variables: 12
- solutions: 1
- calculation time (secs): 600
- edits: 3000

24. Tables 4 and 5 show that the performance of SLICE 1.6 is about the same for missing and zero values. Variables with missing values are first eliminated from the set of edits, because missing values cannot be fixed. Nonetheless, for records with missing values it happens more often that the number of edits in a node becomes too large. If missing values are set at zero, about one third of the records are already clean for Car trade. Table 5 classifies the computation time and result for each record in test wave 5. The PC we used is a DELL 3 GHz with 1 GB RAM. A substantial part of the records is edited successfully within 2 seconds, but some take more than 30 seconds. Only three records do not result in a solution within ten minutes.

Table 4. SLICE results for different treatments of missing values in test wave 5.

Result	With missing values		Missing values set a zero	
	Records	Percentage	Records	Percentage
Clean	2	1.0%	71	35.0%
Success	178	87.7%	122	60.0%
Success TooManyEditsAtNode	20	9.9%	4	2.0%
Success TakingTooLong	0	0.0%	3	1.5%
TakingTooLong	1	0.5%	1	0.5%
TooManyEditsAtNode	0	0.0%	1	0.5%
TakingTooLong TooManyEditsAtNode	2	1.0%	1	0.5%
Total	203	100.0%	203	100.0%

25. We now discuss the final SLICE input and performance for small and medium-sized enterprises for Car trade, Wholesale trade, and Retail trade, for test waves 6 and 7. Within a branch, we often have several *domains of observation (DoO)*. For instance, for Car trade we have ten domains of observation; five for small enterprises and five for large and medium-sized enterprises. Namely, Trade and repair of cars (I+II), Trade in car parts, Trade in and repair of motorcycles (parts), and service stations. The

number of variables can differ per domain of observation. One reason for the large number of edits per trade \times size class is that we have many edits of the form ‘IF *DoO* = value THEN subtraction/addition rule’. These edit rules do not affect the performance of SLICE if they are not relevant for a specific *DoO*. In table 6, we therefore give the number of edits for a specific *DoO*.

Table 5. Classification of computation time and result of SLICE in test wave 5.

		Results					Total
		Clean	Success	Success TooManyEdits AtNode	Success TakingTooLong	TakingTooLong and/or TooManyEdits- AtNode	
Missing values	$t \leq 2$	2	114				116
	$2 < t \leq 6$		32				32
	$6 < t \leq 30$		23	7			30
	$30 < t \leq 60$		4	4			8
	$60 < t \leq 180$		5	3			8
	$180 < t \leq 600$			6			6
	$600 < t$					3	3
	Total		178	20		3	203
Filled in zeros	$t \leq 2$	71	56				127
	$2 < t \leq 6$		30				30
	$6 < t \leq 30$		26				26
	$30 < t \leq 60$		3	2			5
	$60 < t \leq 180$		5	1			6
	$180 < t \leq 600$		2	1	3		6
	$600 < t$					3	3
	Total	71	122	4	3	3	203

26. Table 6 shows that the number of variables for small enterprises slightly increased for SBS 2006. For larger enterprises, there are at least 20 variables less for SBS 2006, compared with SBS 2005. The number of edit rules for SLICE is often much larger for 2006. However, the number of edits for a specific domain of observation is smaller for medium-sized enterprises. The pass rate for 2006 is still smaller for these enterprises, because of the increased complexity of edit rules and the protection of values of variables.

27. In test wave 6, the main goal is to optimise SLICE. This means walking a tight line between the percentage of records solved (pass rate) and the quality of the solutions. During test wave 6, many edit rules are added to increase outcome quality. This results in a steadily decreasing pass rate. The effect on the pass rate depends on the nature of the new edit rules. We discuss one significant problem below.

28. SLICE has a hard time solving ratio edit rules which aim to force SLICE to transport part of the denominator to the numerator. For instance, we consider the ratio of social costs and gross earnings. SLICE should increase social costs and decrease gross earnings if social costs are less than ten percent of the gross earnings. This is very hard to realise, because of the Fellegi-Holt principle used. If a given edit rule is violated, SLICE tries to solve it by indicating as few erroneous values as possible. In a two variable ratio rule, either the numerator or denominator is often declared correct. We have to formulate several related edit rules to obtain the desired solution. It would be better to solve such systematic errors before starting SLICE, using a deterministic approach.

Table 6. Number of variables, SLICE edit rules and SLICE pass rate, for SBS 2005 and SBS 2006 in test wave 7.

Trade	Year	Small enterprises				Medium-sized enterprises			
		Variables	Edits	Edits for a DoO	Pass rate	Variables	Edits	Edits for a DoO	Pass rate
Car trade	2005	58	96	96	100%	108	167	167	100%
	2006	65-68	172	132	90%	83-85	188	148	91%
Wholesale trade	2005	65	113	112	94%	110	171	170	97%
	2006	69	321	112-113	95%	87-90	345	136	91%
Retail trade	2005	64-65	114	113	100%	110	170	169	98%
	2006	69-70	329	118-119	81%	85-87	352	158	76%

30. In order for SLICE to create optimal results, some edit rules were added or defined differently. Protection status values were also changed and some edit rules were dropped in favour of a better pass rate. Because of undesirable solutions, test wave 6 took up a lot of time and induced many changes in the edit rules and protection status values. These changes caused a big difference between the edit rules used in test wave 5 and test wave 7. Generally, the pass rate of SLICE is lower in the production environment. For instance, the pass rate for medium-sized enterprises for Car trade dropped from 98.5% to 91%. This is also due to the fact that SLICE has only 90 seconds to solve a record in the production environment.

V. MIXED MODE EFFECTS

31. An important question is whether there are mixed mode effects for the editing process. We looked for differences between filled-in paper and electronic questionnaires for four trades in test wave 7, namely construction industry, car trade, wholesale trade, and retail trade. In particular for differences in:

- the trade and size class for enterprises that filled in a paper or electronic questionnaire;
- the number of hard edit failures and missing values per type of questionnaire;
- the number of paper and electronic questionnaires that are considered plausible enough for automatic editing;
- the pass rate of SLICE for paper and electronic questionnaires.

32. Table 7 shows that for construction industry, car trade, and wholesale trade the percentage of electronic questionnaires is relatively small for small enterprises. However, a large part of the enterprises fill in an electronic questionnaire, especially wholesale trade. Electronic questionnaires on average contain less hard edit failures, because subtotals and entries in the profit-and-loss account are filled in automatically when a respondent fills in a questionnaire. For the same reason, the percentage of missing values is smaller for electronic questionnaires (see Table 8). This percentage is still quite large (27% - 35%), because several variables are hardly filled in by the respondent.

33. Table 9 shows that the type of questionnaire has a considerable effect on the percentage of records that are selected for automatic editing. Filled-in paper questionnaires are sent to SLICE relatively more often. Plausibility indicators for SBS 2006 do not consider the number of missing values and edit failures. However, we did not expect that filled-in electronic questionnaires would be considered less plausible by our plausibility indicators. We need to investigate whether respondents are less accurate when filling in the non-automatic entries in an electronic questionnaire. An interesting mixed mode effect for SLICE is shown in Table 10. The pass rate for filled-in electronic questionnaires is higher than the pass rate for filled-in paper questionnaires. This is partly due to the automatic fill-in procedures in electronic questionnaires.

Table 7. Percentage of paper and electronic questionnaires per trade and size class.

Trade	Questionnaire	Size class			Total Percentage	Total Number
		Small	Medium	Large		
Construction industry	Paper	27.2%	19.6%	16.0%	21.7%	907
	Electronic	72.8%	80.4%	84.0%	78.3%	3275
Car trade	Paper	24.8%	17.0%	15.5%	21.2%	398
	Electronic	75.2%	83.0%	84.5%	78.8%	1481
Wholesale trade	Paper	17.1%	14.2%	8.8%	14.5%	889
	Electronic	82.9%	85.8%	91.2%	85.5%	5243
Retail trade	Paper	22.5%	24.6%	15.3%	22.8%	494
	Electronic	77.5%	75.4%	84.7%	77.2%	1677

Table 8. Average percentage of missing values per trade and size class.

Trade	Questionnaire	Size		
		Small	Medium	Large
Construction industry	Paper	56.8%	48.3%	40.0%
	Electronic	34.3%	34.5%	27.1%
Car trade	Paper	49.8%	44.4%	46.0%
	Electronic	34.2%	34.4%	31.9%
Wholesale trade	Paper	49.6%	44.8%	40.2%
	Electronic	34.0%	33.9%	29.4%
Retail trade	Paper	47.9%	45.3%	38.9%
	Electronic	29.7%	34.7%	29.2%

Table 9. Percentage of paper questionnaires for SLICE, per trade and size class.

Trade	For SLICE	Size class		
		Small	Medium	Total Percentage
Construction industry	No	24.8%	19.0%	20.9%
	Yes	32.8%	22.0%	26.6%
Car trade	No	25.3%	14.4%	19.6%
	Yes	24.0%	21.7%	23.3%
Wholesale trade	No	11.7%	8.6%	9.4%
	Yes	25.8%	24.5%	24.9%
Retail trade	No	14.8%	19.3%	17.4%
	Yes	32.2%	35.5%	35.4%

Table 10. Pass rate for SLICE records and differences between electronic and paper questionnaires.

Trade	Medium	Small enterprises		Medium-sized enterprises	
		SLICE records	Pass rate	SLICE records	Pass rate
Car trade	Electronic	445	93%	206	92%
	Paper	144	83%	57	89%
Wholesale trade	Electronic	423	96%	1115	91%
	Paper	147	90%	362	90%
Retail trade	Electronic	206	87%	229	76%
	Paper	126	75%	112	68%

VI. CONCLUSIONS

34. At Statistics Netherlands, SBS questionnaires for 2006 were redesigned to decrease the response burden. This had a lot of consequences for the selective editing process. Plausibility indicators and the automatic editing program SLICE use information from last year. The redesign meant that some reference values for variables for year t could not be derived from variables for year $t-1$. We therefore

used manually edited data for year t for automatic editing of year t . For some important variables, we approximated reference values for plausibility indicators using related variables for year $t-1$.

35. The partial transition from paper to electronic questionnaires gave less missing values and errors to be solved. However, it had a negative effect on the number of records that is selected for automatic editing. That is, electronic questionnaires are considered less plausible by our plausibility indicators. They do result in a better pass rate for SLICE compared to paper questionnaires. We need to investigate whether respondents are less accurate when filling in the non-automatic entries in an electronic questionnaire.

36. Adapting SLICE to new SBS-questionnaires and making up arrears have lead to some significant changes for SLICE, the main one being the integration of SLICE edit rules with BLAISE edit rules for interactive editing. Another difference is the treatment of missing values, which can now differ for each variable. In most cases, missing values are imputed, but sometimes they are treated as zeros. Due to these changes, SLICE comes up with more plausible solutions. However, the pass rates of SLICE can be lower.

References

- Fellegi, I., and D. Holt, 1976, *A systematic approach to automatic edit and imputation*. Journal of the American Statistical Association, Vol. 71, pp. 17-35.
- Giesen, D. and T. Hak, 2005, *Revising the Structural Business Survey: From a MultiMethod Evaluation to Design*. Paper presented at the Federal Committee on Statistical Methodology Research Conferences, Arlington, Virginia, November 14-16 2005.
- Granquist, L. and J. Kovar, 1997, Editing of Survey Data: How Much is Enough? In: *Survey Measurement and Process Quality* (ed. Lyberg, Biemer, Collins, De Leeuw, Dippo, Schwartz, and Trewin), John Wiley & Sons, pp. 415-435.
- Hoogland, J., 2006, Selective editing using Plausibility Indicators and SLICE. In: *Statistical Data Editing Volume No. 3, Quality*. United Nations, New York and Geneva.
- Jong, A., 2002, UniEdit: Standardised processing of structural business statistics in The Netherlands. Invited Paper for UNECE Work Session on Statistical Data Editing, 27-29 May 2002, Helsinki.
- Quere, R., 2000, *Automatic editing of numerical data*. Research paper 0016, Statistics Netherlands, Voorburg.
- Waal, T. de, 2000, SLICE: generalised software for statistical data editing and imputation. In: *Proceedings in computational statistics 2000* (ed. J.G. Bethlehem and P.G.M. van der Heijden), Physica-Verlag, Heidelberg, pp. 277-282.
- Waal, T. de, 2002, *Algorithms for automatic error localisation and modification*. Report BPA-no 901-02-TMO, Statistics Netherlands, Voorburg.
- Waal, T. de, and R. Quere, 2003, A Fast and Simple Algorithm for Automatic Editing of Mixed Data. Journal of Official Statistics 19, pp. 383-402.