WP.15
ENGLISH ONLY

UNITED NATIONS STATISTICAL COMMISSION and
ECONOMIC COMMISSION FOR EUROPE

CONFERENCE OF EUROPEAN STATISTICIANS

Work Session on Statistical Data Editing
(Neuchâtel, Switzerland, 5-7 October 2009)

Topic (iii): Editing and imputation of administrative and census data

ENHANCEMENTS TO THE 2011 CANADIAN CENSUS E&I SYSTEM

Invited Paper

Prepared by Michael Bankier and Sean Crowe, Statistics Canada

I. INTRODUCTION

1. Since the 1996 Census, NIM (Nearest-Neighbour Imputation Methodology) and then CANCEIS (Canadian Census Edit and Imputation System) have been used to perform minimum change donor edit and imputation (E&I) and then later deterministic imputation for progressively more variables in the Canadian Census of Population. By the 2006 Census, CANCEIS was used to do E&I for all census variables. See Bankier (2009) for more information on the development of these systems.

2. The minimum change donor methodology developed by Bankier (1993, 1999) for NIM and CANCEIS identifies the nearest neighbour donors first and then determines the minimum number of variables to impute for a failed record/donor pair. This contrasts with the approach of Fellegi and Holt (1976) that determines the minimum number of variables to impute first and then finds nearest neighbour donors. Reversing the order of operations under Bankier's approach results in an imputation process that is more data driven. Since it is also more efficient computationally, this allows for the imputation of numeric and qualitative data simultaneously.

3. A prototype of CANCEIS was used in the 2000 Brazilian and Swiss population censuses (and will be used in their 2010 population censuses as well). CANCEIS was used in the 2005 Peruvian Census of Population, the Brazilian 2006/2007 Census of Agriculture and the 2009 New Zealand General Social Survey. CANCEIS is used annually with the Statistics Canada Survey of Household Spending. It will also be used in the 2010 Brazilian and the 2011 UK population censuses.

4. For the 2011 Census, CANCEIS is being rewritten in the programming language C# to make it easier to debug and to maintain. In this paper, some enhancements appearing in this new version of CANCEIS will be described.

5. Section II outlines the CANCEIS methodology. Sections III, IV, V and VI discuss some enhancements to CANCEIS for 2011. Section III describes how CANCEIS ensures that outlier responses are neither increased or decreased excessively by imputation. Section III also explains how failed units can be used as donors subject to certain checks and controls. Section IV shows how the user can require that donors match the failed unit very closely on all the matching variables or failing that, a subset. Section V explains how the weight and imputability of a variable can be changed for units or subunits with specific characteristics to help influence which variables will be imputed. Section VI looks at how the use of donors can be optimized without their being overused. Section VII provides a few concluding remarks.

II. BRIEF DESCRIPTION OF CANCEIS METHODOLOGY

6. Bankier (1999) provides an overview of the minimum change donor methodology used by CANCEIS. Some additional details are given in this section. CANCEIS processes units[1] (e.g. a household or family) which may be made up of subunits (e.g. persons within a household).

7. The edits are defined using Derive or Primary Edit Decision Logic Tables (DLTs). Table 1 of Appendix A is a Derive DLT while Table 2 is a Primary Edit DLT. These are two of the tables used to perform E&I on 2006 Census personal income variables. See Section III for a more detailed discussion of the Primary Edits.

8. A Derive DLT is used to derive new variables and do deterministic imputation. In Table 1, the Derive DLT is subdivided into the following sections: DLT Parameters, Common Actions, Deterministic Edits and Deterministic Edit Actions. The conditions (also sometimes called propositions) of the Deterministic Edits can include both numeric and qualitative variables. Each Deterministic Edit Action in Table 1 causes another Derive DLT to be executed but could instead derive a new variable or perform a deterministic imputation.

9. A Primary Edit DLT determines which units pass or fail the edits prior to donor imputation. A unit fails the primary edits if it has an invalid or blank response or if it matches an edit which flags some inconsistency in the responses between two or more variables. No Deterministic Edit Actions are defined in a Primary Edit DLT since CANCEIS determines the minimum number of variables to impute for a failed unit and borrows the responses to impute from a donor. Table 2 show that a condition can be made up of the linear combination of many numeric variables.

10. Note that the conditions of Table 1 and 2 use classes such as WEEKS = CLASS(W49_W52). A class represents a set of possible responses. Thus WEEKS = CLASS(W49_W52) is true if WEEKS equals 49, 50, 51 or 52.

11. For the rest of this paper, the focus will be on CANCEIS donor imputation rather than CANCEIS deterministic imputation. The primary edits are defined using one or more Primary Edit DLTs. The condition result vector lists for each condition in a DLT if it is true (Y) or false (N) for the unit being edited. A primary edit fails if its Ys and Ns match the Ys and Ns in the condition result vector (with any condition with a blank for a primary edit being ignored).

12. CANCEIS finds a number of units (called nearest neighbours or donors for short) that are closest to the failed unit in terms of a distance measure. These donors are used to generate imputation actions. Let $\underline{V}_f = [V_{fi}]$ and $\underline{V}_p = [V_{pi}]$ be I x 1 vectors containing the I variables (either unit or subunit level variables) that enter the distance measure for the failed unit and the donor unit, respectively. The L_p norm distance measure is

$$D_{rfp} = D_r(\underline{V}_f, \underline{V}_p) = \left(\sum_{i=1}^{I} w_i D_i^r(V_{fi}, V_{pi})\right)^{\frac{1}{r}}$$

where r ≥ 1 (r may be non-integer) is an exponent and $D_i(V_{fi}, V_{pi})$ is a distance function giving the distance between the response of the failed unit (V_{fi}) and the response of the donor unit (V_{pi}) for the ith variable. When r = 2 and $D_i(V_{fi}, V_{pi}) = |V_{fi} - V_{pi}|$ for i = 1 to I, D_{rfp} is the standard Euclidean distance. Normally exponents would be represented by p and 1/p in the L_p norm. But because the subscript p appears in V_{pi}, the powers r and 1/r are used here instead.

[1] Note that a term will be underlined when it is first used.

13. Quite frequently, particularly with qualitative variables, r = 1 and $D_i(V_{fi}, V_{pi}) = 0$ when $V_{fi} = V_{pi}$ and $D_i(V_{fi}, V_{pi}) = 1$ otherwise. We will assume in general that $0 \leq D_i(V_{fi}, V_{pi}) \leq 1$ for both numeric and qualitative variables unless stated otherwise. A wide variety of distance functions are provided in CANCEIS and different ones can be used with different variables.

14. The weight w_i associated with a variable (which is non-negative) in the distance measure D_{rfp} can be given a larger value if it is believed that the variable is likely to be error free and is a good predictor for the values of other variables often in error. The distance measure can include auxiliary variables which are variables that enter the distance measure but not the edits.

15. To ensure the best donors are selected, the subunits of a failed unit can be reordered in various ways to see which reordering results in the smallest distance for a particular donor unit. Smaller distances may result through reordering because, for example, children can be listed in ascending order based on age in one household and descending order in another household.

16. Only nonmatching variables (those with $V_{fi} \neq V_{pi}$) are, of course, considered for imputation. All subsets of these nonmatching variables are efficiently assessed to determine which are the optimum imputations for a failed unit/donor pair. Each of these subsets, when imputed, will be called an imputation action (IA) and will be represented by $\underline{V}_a$. Those IAs which fail the edits (see Section III) are discarded. Those IAs which pass the edits are retained and are called feasible IAs.

17. For each feasible IA, we calculate

$$D^r_{rfpa} = \alpha D^r_{rfa} + (1-\alpha) D^r_{rap}$$

where $D_{rfa} = D_r(\underline{V}_f, \underline{V}_a)$ and $D_{rap} = D_r(\underline{V}_a, \underline{V}_p)$. D_{rfpa} is a weighted average of the distance D_{rfa} of the IA to the failed unit and the distance D_{rap} of the IA to the donor. Placing an emphasis on minimizing D_{rfa} (by having the parameter α equal 0.75 or 0.90 say) means that CANCEIS will tend to modify the data of $\underline{V}_f$ as little as possible through imputation. Placing some weight on D_{rap}, however, means that some importance is given to having a plausible IA, i.e. one that resembles the donor.

18. For the feasible IAs , the minimum value of D_{rfpa} is determined and is labeled $\min D_{rfpa}$. Any feasible IAs with $D_{rfpa} = \min D_{rfpa}$ will be called minimum change IAs. Those feasible IAs with a D_{rfpa} that satisfies the equation

$$D_{rfpa} \leq \gamma \min D_{rfpa}$$

are called near minimum change imputation actions (NMCIAs) and are retained on a List of NMCIAs where $\gamma \geq 1$ (e.g. 1.1). Values of the parameter γ greater than 1 are allowed because the NMCIAs, for practical purposes (particularly with numeric variables), are nearly as good as the minimum change IAs. IAs, which are not NMCIAs, are discarded because otherwise the principle of making as little change to the data as possible, when carrying out imputation, is being violated.

19. Only NMCIAs which are essentially new (i.e. no subset of the variables being imputed **based on that donor** would pass the edits) are retained. IAs that are not essentially new are discarded because one or more variables are being unnecessarily imputed. Doing this again satisfies the principle of making as little change to the original data as possible. One of the remaining NMCIAs will be randomly selected and retained for that failed unit.

20. In the Canadian Census, variables are split into non-overlapping groups and then these groups are processed sequentially by separate E&I modules. Additional variables can appear in the edits of the module currently being processed, but which were finalized in an earlier E&I module. These variables cannot have their values imputed by the current module and hence are labeled unimputable. If the current

module was allowed to change these unimputable variables, there would be no guarantee that the units would still pass the edits of the earlier modules.

III. SECONDARY EDITS AND USING FAILED UNITS AS DONORS

21. This section describes in detail how CANCEIS determines which units pass and which fail editing during donor imputation, how CANCEIS determines which units are possible donors for a failed unit, and what properties imputed units must have.

22. Nonresponse (for example, blanks) and invalid responses are identified for CANCEIS by the variable's value not being on a list of valid responses (which is called a validity set). For the rest of this report, the term invalid will be taken to refer to both nonresponse and invalid responses.

23. CANCEIS allows two types of edit rules (or edits for short):
- primary edits which define combinations of responses considered inconsistent. If a unit matches one or more primary edits, it is said to fail the primary edits. Otherwise it is said to pass the primary edits. Primary edits are defined in Primary Edits DLTs.
- secondary edits which define combinations of responses considered valid and consistent but rare. These rare combinations of responses are sometimes called outliers. If a unit matches one or more secondary edits, it is said to fail the secondary edits. Otherwise it is said to pass the secondary edits. Secondary edits are defined in Secondary Edit DLTs.

24. If a unit has one or more invalid values or if it fails one or more primary edits, it will be called a failed unit or a unit requiring imputation. Conversely, if the unit has only valid values and passes all the primary edits, it will be called a passed unit.

25. After successful minimum change donor imputation by CANCEIS, a failed unit will be converted into a passed unit (all values are now valid and all primary edits are now passed). Whether some or all of the secondary edits must also be passed depends upon the user-defined parameters that will be discussed below.

26. Secondary edits were introduced in the 1996 Census to avoid having imputation disproportionably increase the number of outliers. Widowed persons aged 15 to 24 passed the primary edits but were few in number before imputation. These will be referred to as young widows. Minimum change donor imputation of the demographic variables initially resulted in a large percentage increase in the number of young widows even though in absolute terms the increase was small. As an example, an 18-year-old with nonresponse to the marital status question could have "widowed" marital status imputed from a 40-year-old donor. In this case, neither the person who failed the edit nor the donor was a young widow but, by mixing the responses of the failed edit person and the donor, a young widow was created through imputation.

27. To avoid this happening, a secondary edit was introduced which did not allow a person in a failed household to be a young widow after imputation. In addition, households containing young widows could not be used as donors during imputation. This, however, had some unintended consequences. For example, if a young widow failed the primary edits because the response for sex was missing then he/she could no longer be a young widow after imputation. For this reason, improvements to how secondary edits are applied will be implemented for the 2011 Census.

28. In the 2006 Census, it was always required that donors passed both the primary and secondary edits. Sometimes this resulted in relatively few donors being available for failed units. For 2011, units which contain some invalid values, or fail one or more of the primary or secondary edits can be used as donors if desired for certain types of failed units.

29. For a given failed unit requiring imputation, paragraphs 30, 31 and 33 below give a series of conditions which will result in certain units being discarded as donors.

30. By default, the set of potential donors is restricted to the units that have valid values for all the variables. By changing the value of a certain system parameter, the set of potential donors is increased by only discarding those with invalid values for the variables where the unit being imputed has invalid values. With this larger set of potential donors, if a donor has invalid responses for some variables, these variables will be set to non-imputable for the failed/donor pair when generating imputation actions in order to avoid imputing invalid responses.

31. With the donors remaining, by default discard any donors that fail one or more primary edits. By changing the value of a certain parameter at the DLT level, it is possible to increase the number of donors available by discarding only those that fail one or more of the edits that the unit being imputed fails.

32. The set of secondary edits is reduced for this particular failed unit and its donor in two steps:
- identify any secondary edits that this failed unit fails and drop them from the secondary edits.
Then, by default,
- identify any secondary edits that this donor unit fails and drop them from the secondary edits
By changing the value a certain parameter at the DLT level, there is the option of retaining some or all of the secondary edits that this donor unit fails. The secondary edits that remain are called the reduced secondary edits. As an example, if a failed subunit is a young widow, the secondary edit forbidding young widows is dropped. This is done so as to not eliminate young widows through imputation. If a donor subunit is a young widow, we have the option of dropping the secondary edit forbidding young widows. Optionally dropping this secondary edit makes sense since if the donor is a young widow, it is not unreasonable to allow another young widow to be created through imputation.

33. With the donors remaining, by default discard any donors that do not pass all the reduced secondary edits. By changing the value of a certain parameter at the DLT level, no additional donors are discarded but we set to unimputable (for this failed unit/donor unit pair) any variables entering a reduced secondary edit in this DLT that this donor unit fails. This means that the donor can be used to impute other variables but not the variables entering that secondary edit. A donor should be immediately discarded, however, if a variable for which the failed unit has an invalid value is set to unimputable at this point. With the same parameter at the DLT level, it can also be specified that no additional donors are to be discarded and no additional variables are to be made unimputable.

34. When generating imputation actions for a failed unit/donor pair, both the primary and reduced secondary edit rules must be passed by an imputation action for it to be retained.

IV. MANDATORY AND SEMI-MANDATORY MATCHING VARIABLES

35. For the mandatory and semi-mandatory matching variables discussed in the following two paragraphs, it is assumed for simplicity that $D_i(V_{fi}, V_{pi}) = 1$ when $V_{fi} \neq V_{pi}$ and $D_i(V_{fi}, V_{pi}) = 0$ when $V_{fi} = V_{pi}$. More complex distance functions can also be used to make this approach more flexible. For example, one function allows $D_i(V_{fi}, V_{pi})$ to be zero or close to it when the values V_{fi} and V_{pi} are not equal but are 'very similar'.

36. Sometimes the user wants all donors to match the failed units on certain mandatory matching variables. To do this, the mandatory matching variables each have w_i in paragraph 12 set equal to 1000 say and then the user sets the parameter 'InitialMaximumDrfp' equal to 999. This instructs CANCEIS that any donor used must have $D^r_{rfp} \leq 999$.

37. Sometimes the user ideally wants donors to match the failed units on as many semi-mandatory matching variables as possible. The user may realize, however, that this may not always be possible and hence allows one (but no more than one) semi-mandatory variable not to match in some cases. To do this, the semi-mandatory variables each have w_i set equal to 1000 say and 'InitialMaximumDrfp' is set to 1999 Thus any donor that does not match on two or more semi-mandatory matching variables is immediately discarded. New for 2011, a second parameter 'OptimalMaximumDrfp' can be set to 999 in

this case. This means that the donor search will continue as long as an 'optimal' donor (matches on all semi-mandatory matching variables, i.e. $D^r_{rfp} \leq 999$) has not been found. If an 'optimal' donor is found, any donors not matching on one semi-mandatory variable, along with their associated NMCIAs for that failed unit, are discarded.

V. ALLOW VARIABLE WEIGHTS AND IMPUTABILITY TO VARY BY SUBUNIT/UNIT

38. New for 2011, there will be an optional Derive DLT in a donor module which will allow the user to modify, for units/subunits with special characteristics, whether a variable is imputable or not as well as its weight w_i. This will allow the user to ensure that some variables are never imputed and that other variables have a higher probability of being imputed. This should improve the quality of the imputation actions where there is a strong indication of what variables should be imputed. In addition, the parameters 'InitialMaximumDrfp' and 'OptimalMaximumDrfp' (as described in Section IV) can also be changed. This is essential when using semi-mandatory variables since some units may have inconsistent responses where one or more of these semi-mandatory variables will *have* to be imputed (and hence *must* have a donor that does not match on one or more of them).

VI. LIMITS ON DONOR USAGE

39. In general, a donor should not be used too often since it will have undue influence on the imputed data and could introduce a spike in the distribution of imputed results. The CANCEIS parameter 'DonorUsageLimit' places an upper limit on how many failed units can use one specific donor. CANCEIS for 2011 will identify those failed units which would have the largest increase in D_{rfpa} if they were denied access to an overused donor. These failed units will then be given priority by CANCEIS to use this donor while still having the 'DonorUsageLimit' limit respected overall. This should encourage the users to set 'DonorUsageLimit' to lower values than were used in 2006 knowing that any increases in D_{rfpa} will be limited.

VII. CONCLUDING REMARKS

40. The enhancements outlined in this paper were identified as important to implement based on our experiences from the 2006 Census. These are a small but important subset of the enhancements planned for CANCEIS for the 2011 Census. These enhancements were chosen based on biweekly meetings of methodology, subject matter and systems staff where the relative merits of each enhancement was debated.

References

Bankier, Mike (1993), "Imputing Numeric and Qualitative Census Variables Simultaneously", Social Survey Methods Division Report, Statistics Canada, Dated April 14, 1993.

Bankier, M. (1999), "Experience with the New Imputation Methodology Used in the 1996 Canadian Census with Extensions for Future Censuses", Working Paper 24, Proceedings of the UN/ECE Work Session on Statistical Data Editing, Italy (Rome).
(http://www.unece.org/stats/documents/1999.06.sde.htm)

Bankier, M. (2009), "Evolution of Canadian Census E&I Systems – 1976 to 2011", Working Paper 22, Proceedings of the UN/ECE Work Session on Statistical Data Editing, Switzerland (Neuchâtel).

Fellegi, I.P. and Holt, D. (1976), "A Systematic Approach to Automatic Edit and Imputation", Journal of the American Statistical Association, March 1976, Volume 71, No. 353, 17-35.

Appendix A: Examples of CANCEIS Decision Logic Tables from the 2006 Census

Table 1: Example of Derive Decision Logic Table

```
% DLT Name:                  IND3                                    |DLT Parameters
                                                                     |
% Strata:                    1                                       |
% Purpose:                   DERIVE                                  |
% Type:                      CONFLICT                                |
% Overlapping:               NO                                      |

% Symmetry:                  NO                                      |


$ DECL (TEICBN,VINCPOS)                                              |Common Actions
$ DECL (TGOVTI,VINCPOS)                                              |Performed on all units
*                                                                    |Derive DLT Only
$ TEICBN  = EICBNF                                                   |
$ TGOVTI  = GOVTIF                                                   |
$ STINI5  = 0                                                        |
$ FSTINI5 = 0                                                        |
$ INI5IMP = FALSE                                                    |
$ INCDONOR5 = NOT_TOUCHED                                            |
$ DO CLEAN_INI1_INI4                                                 |
$ DO FIX_INI1_INI4
*|Conditions                 Columns Represent Edit Rules            |Deterministic Edits
@ DOCTP = HHLD_2C            ;Y;N;N;N;N;N;N;N;N;N;N;N;N;ELSE;        |Note that both
@ RRC   = RC1                ; ;Y;N;N;N;N;N;N;N;N;N;N; ;             |qualitative (SEX=MALE) and
@ RRC   = RC2                ; ; ;Y;N;N;N;N;N;N;N;N;N; ;             |numeric (AGE<20) variables
@ AGE   < 20                 ; ; ; ;Y;Y;Y;N;N;N;N;N;N; ;            |appear in the conditions.
@ PRCODE = QUEBEC            ; ; ; ;Y;Y;N; ; ; ; ; ; ; ;             |
@ SEX   = MALE               ; ; ; ;Y;N; ; ; ; ; ; ; ; ;             |Edit is failed if for all
@ AGE   > 65                 ; ; ; ; ; ; ;Y;N;N;N;N;N; ;             |Y/Ns in edit, condition is
@ LSTWK = NEVER              ; ; ; ; ; ; ; ;Y;N;N;N;N; ;             |true where there is a Y and
@ EICOW = UFW                ; ; ; ; ; ; ; ; ;Y;N;N;N; ;             |false where there is a N.
@ EICOW = OTH                ; ; ; ; ; ; ; ; ; ;Y;N;N;N; ;           |
@ SRC7  = PARTIAL            ; ; ; ;Y; ; ; ; ;N;N; ; ; ; ;           |
@ TEICBN > 0                 ; ; ; ;N; ; ; ; ;N;N; ; ; ; ;           |
@ WEEKS = CLASS(W49_W52)     ; ; ; ; ; ; ; ; ; ; ;Y;N;N; ;           |
@ WEEKS = CLASS(W40_W48)     ; ; ; ; ; ; ; ; ; ; ; ;Y;Y; ;           |
@ EICOW = SENF               ; ; ; ; ; ; ; ; ; ; ; ;Y;N; ;           |
@ EICOW = SEF                ; ; ; ; ; ; ; ; ; ; ; ; ;Y; ;
*|Conditional Actions        Columns show actions to do              |Deterministic Edit Actions
& DO IND3B                   ; ; ; ;X; ; ; ; ; ; ; ; ;X;             |If edit failed, any action
& DO SET_AGEDEC31            ;X;X;X;X;X;X;X;X;X;X;X;X;X;X;           |with X beside it for that
                                                                     |edit is carried out.
```

Table 2: Example of Primary Edit Decision Logic Table

```
% DLT NAME:                  INI1_TOTCONSIS
*
% STRATA:                    1-3
% PURPOSE:                   PRIMARY EDITS
% TYPE:                      CONFLICT
*
@ WAGESF=CLASS(MISSING_INC)                                                                                    ;N;N;
@ FARMIF=CLASS(MISSING_INC)                                                                                    ;N;N;
@ SELFIF=CLASS(MISSING_INC)                                                                                    ;N;N;
@ OASGIF=CLASS(MISSING_INC)                                                                                    ;N;N;
@ CQPPBF=CLASS(MISSING_INC)                                                                                    ;N;N;
@ EICBNF=CLASS(MISSING_INC)                                                                                    ;N;N;
@ GOVTIF=CLASS(MISSING_INC)                                                                                    ;N;N;
@ INVSTF=CLASS(MISSING_INC)                                                                                    ;N;N;
@ RETIRF=CLASS(MISSING_INC)                                                                                    ;N;N;
@ OTINCF=CLASS(MISSING_INC)                                                                                    ;N;N;
@ WAGESF+FARMIF+SELFIF+OASGIF+CQPPBF+EICBNF+GOVTIF+INVSTF+RETIRF+OTINCF>INI1_MAX_10                            ;Y; ;
@ WAGESF+FARMIF+SELFIF+OASGIF+CQPPBF+EICBNF+GOVTIF+INVSTF+RETIRF+OTINCF<INI1_MIN_10                            ; ;Y;
```