Survey Methodology, June 1985
Vol. 11, No. 1, pp. 79–88
Statistics Canada

1981 Census of Agriculture Data Processing Methodology

DAVID K. HOLLINS[1]

ABSTRACT

This paper presents an overview of the methodology used in the processing of the 1981 Census of Agriculture data. The edit and imputation techniques are stressed, with emphasis on the multivariate search algorithm. A brief evaluation of the system's performance is given.

KEY WORDS: Edit and imputation; Multivariable searches

1. INTRODUCTION

This paper presents an overview of the methodology used in the processing of the 1981 Census of Agriculture data. There are 3 separate phases to the processing of the data: Data Entry, Edit, and Imputation, each of which performs a different function. First, in Data Entry, data on the questionnaires are keyed onto a computer data file. Then, in the Edit phase, computer edits are applied to the keyed data records in order to detect any inconsistent, missing, or suspicious entries. In the final phase, Imputation, actions are taken to adjust the data records so that they conform to the rules defined by the computer edits applied during Edit. The methodology involved in each of the three phases of processing is described in subsequent sections of this paper. A flow chart of the 1981 Census of Agriculture processing is given in Figure 1.

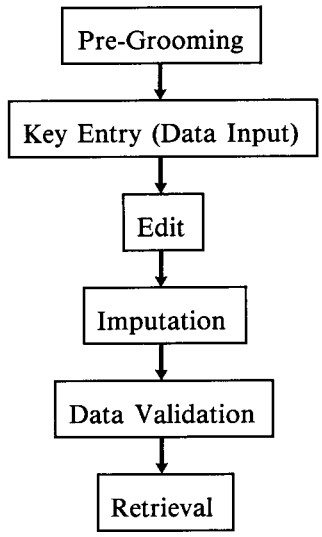

Figure 1. Overall Process Flow

[1] D.K. Hollins, Census and Household Survey Methods Division, Statistics Canada, Tunney's Pasture, Ottawa, Ontario, Canada K1A 0T6.

The 1981 Census of Agriculture required that the same questionnaire be completed by each farm operator in Canada. The questionnaire is 8 pages long and consists of 134 questions. Questions are asked on all aspects of farm operation, including items such as types of crops grown, livestock raised, equipment maintained, and types of land use. Operators are required to answer only those sections of the questionnaire which apply to their holding.

As this paper is an overview, it is not possible to delve into the technical computer aspects of the Census of Agriculture processing. These details may be found in Shields and Yiptong (1981), on which this paper is based.

2. DATA ENTRY

In the Data Entry phase the Census of Agriculture data are transferred from the original questionnaires to a data file in computer memory. Data entry is comprised of two stages: a clerical pre-grooming process (Pre-Scan), and Key Entry.

After the questionnaires arrive at head office for processing, a clerical pre-grooming process known as Pre-Scan is performed. In this process, a clerk scans each questionnaire for response irregularities such as unreadable entries, ditto marks, and responses in incorrect locations. If valid responses can be discerned, they are recorded in the appropriate locations, if not, the questionnaire is left unchanged.

Next, in Key Entry, the data on each questionnaire are keyed into the computer. Identifying information from the front page of the questionnaire is entered in a standard fixed format. However, since farm operators are required to answer only the sections of the questionnaire that apply to their holding, a large portion of the questionnaire remains blank. To reduce keying time, a method known as "string-keying" is used to enter the remaining data. This means that the field name is keyed, immediately followed by the data value for that field. Only fields with existing data values are keyed; unanswered portions of the questionnaire are not. Because of the sparseness of the data, this method results in significant savings in keying time required.

The Key Entry process creates one Edit and Imputation Master File (EIMF) record for each of a total of approximately 320,000 questionnaires. There are 244 fields on an EIMF record, each identified by a name, generally 6 characters in length. The Key Entry operator is instructed to key "#" for any unreadable entries. If possible, a clerical correction will be performed on records containing this symbol during Edit, otherwise, the records will be corrected during imputation.

3. EDIT

The Edit phase serves two purposes. The first is to use computer edits to detect any inconsistent, missing, or suspicious entries in the data. The second is to perform a clerical correction on the defective records, or if that is not possible, then to pass the defective records on to be fixed during Imputation. A flow chart of the Edit process is given in Figure 2.

There are 3 components to the edit system: two computer edit cycles called Correction Cycles #1 and #2, and a cycle for correcting edit failures, called Correction of Rejects. Correction Cycle #1 (CC #1) consists of those edits that detect conditions that prevent the "de-stringing" (the conversion from string format to fixed format) of the keyed record (decode edits), and those edits that detect errors in the geographic and identifying information from the front page of the questionnaire (ID edits). Correction Cycle #2 (CC #2) consists of those edits that identify inconsistencies in the main body of the data (data edits). Correction of Rejects is a clerical process during which both CC #1 and CC #2 edit failures are corrected manually. Edit failures that cannot be corrected by Correction of Rejects are passed on to Imputation.

Each of the EIMF records is processed through the edit system individually.

3.1 Correction Cycle #1 (Decode and ID Edits)

Correction Cycle #1 consists of the application and resolution of two sets of edits: the decode edits and the ID edits.

The decode edits are applied first and if conditions exist that prevent the "de-stringing" of the data record, then decode edit failures will result. For example, as no two fields should have the same identifying characters, "de-stringing" will be prevented if two field names are keyed identically.

Any failed decode edits are resolved manually by the Correction of Rejects staff. This involves returning to the questionnaire to determine the cause of the edit failure, then the rekeying of the relevant data. After an attempt is made to resolve a decode edit failure, the EIMF record is re-edited by passing it through the decode edits again, forming a continuous cycle between the decode edits and the Correction of Rejects staff. This cycle is repeated until there

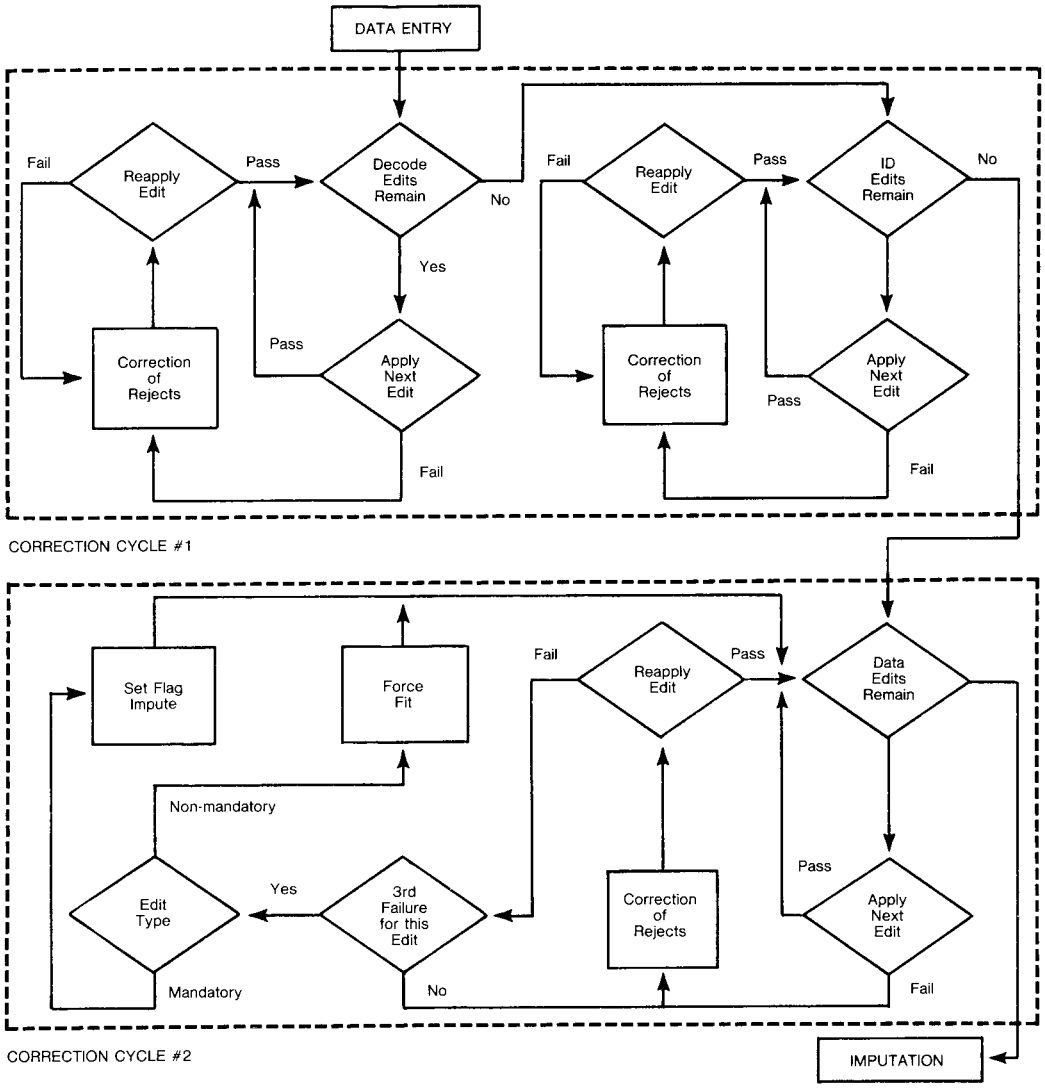

Figure 2. Edit Process Flow

are no decode edit failures remaining on the EIMF record. If a decode edit cannot be resolved directly, the most appropriate valid interpretation of the available data is employed as a final override.

After all decode edit failures have been resolved, the ID edits are applied. If any of the identifying information on the EIMF record is inconsistent or missing, then one or more ID edits will fail. These ID edit failures are resolved in an identical manner to the decode edits.

Once all of the CC #1 (decode and ID) edit failures have been resolved by the Correction of Rejects staff, the EIMF record is passed through the CC #2 edit program.

3.2 Correction Cycle #2 (Data Edits)

The data edits (CC #2) are used to detect errors in the main body of the questionnaire, as opposed to errors in coding, or in identifying information. There are two types of data edits: non-mandatory edits (75), and mandatory edits (24).

Non-mandatory edits are written to detect suspicious entries on the EIMF data records. Generally, non-mandatory edits, detecting variable values falling outside prescribed limits, are performed by comparing different fields or groups of fields on the questionnaire to determine if some data values are abnormally high or low in comparison with others. For example, a record with total farm area equalling 10 acres and containing 10,000 cattle would be flagged by a non-mandatory limit edit.

Mandatory edits are written to detect logical impossibilities on the data record, e.g., if the total number of cattle reported is not equal to the sum of the reported values for each of the different cattle types, then a mandatory edit would fail. The most complex mandatory edits are those written for the crop section of the questionnaire.

To resolve a non-mandatory edit failure, the record is sent to a Correction of Rejects clerk. The Correction of Rejects clerk first notes whether or not the edit failure is due to a keying error. If it is, the relevant data is rekeyed. If it is not, the clerk scans the questionnaire to see if the respondent has written any comments on the questionnaire that may explain the reason for the edit failure. For example, if the respondent is instructed to answer a question in tons, and tons has been crossed out and pounds written in, the response will probably fail a non-mandatory limit edit. In this case, the Correction of Rejects clerk will convert the response from pounds into tons. If the Correction of Rejects clerk can find no explanation for the edit failure, the respondent's answers are left intact on the EIMF record and are indicated acceptable. Although no changes are made to the data on the EIMF record, this is known as "force-fitting" the data.

Mandatory edit failures are handled somewhat differently to non-mandatory edit failures. To resolve a mandatory edit failure, the failed record is sent to a Correction of Rejects clerk who proceeds at first in an identical manner to that used in the resolution of non-mandatory edit failures. However, if no explanation for the edit failure can be found, instead of "force-fitting" the edit failure, the record is flagged for computer imputation.

As in CC #1, there is a continuous cycle between the Correction of Rejects staff and the CC #2 edit program. After each attempt is made to resolve a CC #2 edit failure the EIMF record is re-run through the CC #2 edit program. Unlike CC #1, however, the Correction of Rejects clerk has only 3 attempts to resolve the CC #2 edit failures on a given EIMF record. After the third attempt, the CC #2 edit program is run once again. Any remaining non-mandatory edit failures are marked "force fit" and any remaining mandatory edit failures are marked "impute". The mandatory edit failures are simply flagged at this stage. The particular fields requiring imputation are identified at the imputation stage.

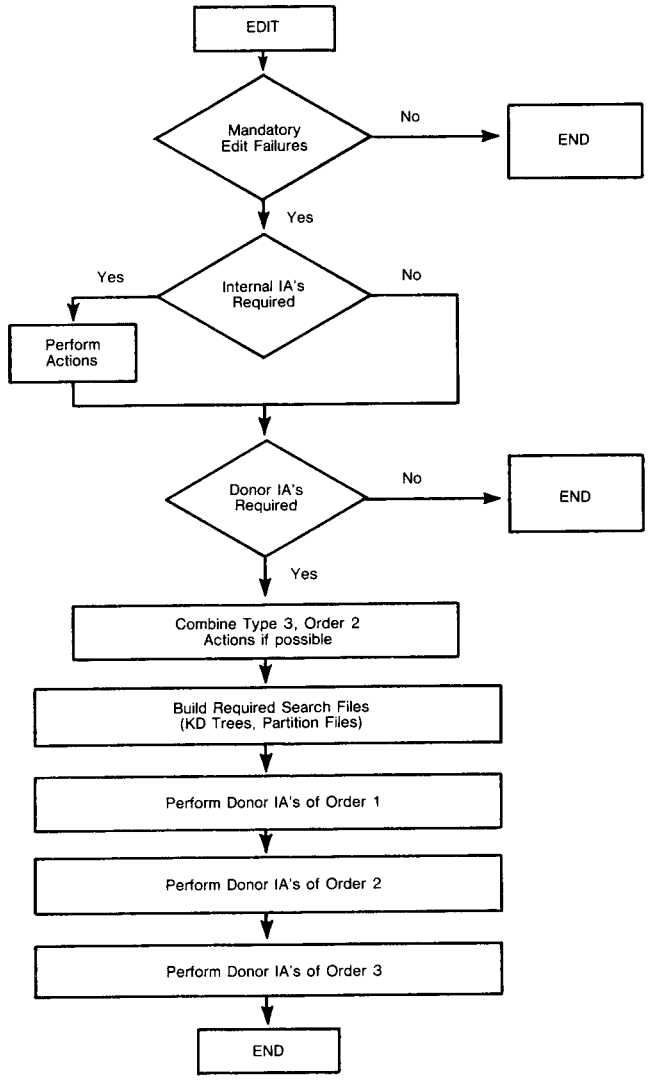

Figure 3. Imputation Process Flow

4. IMPUTATION

The purpose of the 1981 Census of Agriculture imputation system (see Figure 3) is to resolve edit failures on the EIMF data records. As all non-mandatory edit failures are "force-fit" as described in the previous section, only the mandatory edit failures remain to be resolved by the imputation system. In order to make the EIMF data records conform to the mandatory edits, specified "imputation actions" are performed. These imputation actions (IA's), of which there are over 100, are designed so that as few fields as possible are changed on the EIMF record, e.g. totals are always adjusted to equal the sum of the parts, rather than the parts being adjusted to total the sum. Each IA has associated with it the appropriate imputation processing control information and is selected based on the field or fields requiring imputation. There are two different types of IA's performed: internal IA's, or deterministic corrections, and donor IA's.

4.1 Internal Imputation Actions

Internal IA's are performed in cases where sufficient data exists on the failed record to enable the imputation system to provide a deterministic correction for the inconsistent field(s). These internal IA's are performed in cases where the inconsistent field(s) is (are) deterministically dependent on other fields not requiring imputation. For example, an internal IA would be performed if a respondent reports quantities for the various types of cattle but neglects to report the total number of cattle. In this case, total cattle would be calculated using the sum of the quantities reported for the various types of cattle. Another situation in which an internal IA would be performed is where a respondent reports a certain quantity of a particular type of fruit tree but neglects to give the corresponding acreage. In this case, the acreage would be computed using a predetermined average density for that type of fruit tree. Internal IA's are performed in accordance with constraints to ensure that the imputed values are within reasonable bounds.

The implementation of internal IA's is more straightforward than that of donor IA's. As the internal IA is performed using data from the same record, there is no need to specify an algorithm for donor selection. The only requirement is to perform the deterministic correction specified by the appropriate internal IA. All internal IA's are performed before proceeding to donor imputation.

4.2 Donor Imputation Actions

When the inconsistent field or fields are not deterministically dependent on other consistent fields, internal IA's cannot be applied. The lack of sufficient information on the failed record to provide a deterministic correction to the inconsistent field(s) necessitates an imputation method using data contained on another record. This method, known as donor imputation, involves the transfer of data from a "clean" donor record (one which has passed all mandatory edits) to the failed record. The transferred data will restore consistency to the inconsistent field(s) on the failed record. For example, a donor IA will be performed in order to estimate the distribution for types of cattle when only the total number of cattle is reported. In this case, the distribution of cattle types present on the donor record is transferred to the failed (recipient) record.

As donor imputation requires an algorithm for locating a donor record, it is more complex to implement than internal imputation. In order to perform donor imputation, several search "parameters" must be specified.

To ensure that a "clean" donor record is geographically close to the "bad" recipient record, the country is divided into distinct geographical regions called imputation regions. The delineation of these imputation regions is based on the existing "crop district" boundaries which are defined according to characteristics such as soil type and climate. There are 59 crop districts, and thus 59 imputation regions, in Canada with an average of 5,500 farms per region. In order to be an eligible donor, a record must be in the same imputation region as the recipient record.

In order to avoid searching records that cannot donate suitable data, each donor IA also specifies the subpopulation on which the donor search is to take place. For example, if the distribution for types of cattle is being imputed, then the only records searched in order to find a donor would be members of the subpopulation where cattle have been reported. A given record may be a member of several of the 30 different subpopulations. In some cases, all clean records within the imputation region are deemed suitable donors in which case the general population in the imputation region is defined as the appropriate subpopulation.

The final constraint on the file of eligible donors is the fact that records requiring any donor imputation themselves cannot be used as donors. However, records requiring only internal imputation may be used as donors.

In summary, the file of eligible donors consists of all records not requiring donor imputation that are members of the subpopulation specified by the imputation action to be performed and that are also located in the same imputation region as the bad record.

As some records require more than one IA to be performed, there is need for a hierarchical system of imputation action execution. To specify the order in which the IA's are to be performed, every IA, both internal and donor, has one of three "orders" associated with it. IA's of order 1 are performed first, followed by IA's of orders 2 and 3 respectively.

To aid in the selection of a suitable donor record, one or more variables not requiring imputation are selected to be used as matching variables for each donor IA. These matching variables, selected by subject matter experts, are considered to be highly correlated with the field(s) requiring imputation. Both the recipient and the selected donor record should have similar matching variable values. As the use of continuous matching variables does not permit exact matches, a distance function based on the selected matching variable(s) is used to identify the closest eligible donor to the bad record.

Each donor IA has one of three possible search types associated with it. Partition searches (type 1) are performed when only 1 discrete matching variable is specified for the IA. Binary searches (type 2) are performed when only 1 continuous matching variable is specified for the IA. Multivariable searches (type 3) are performed when 2 or more continuous matching variables are specified for the IA. Each of these three search types is described individually in the following sections. Other combinations of matching variable types are not employed.

Finally, after a suitable donor has been selected and if specified in the IA control information, the donated data from the donor record are prorated before transferring them to the recipient record. For example, if the variable "number of trucks" is used as a matching variable for imputing "value of trucks", then the value of "value of trucks" assigned to the recipient record is equal to "value of trucks" of the donor, multiplied by the ratio "number of trucks" of the recipient divided by "number of trucks" of the donor.

As previously described, each donor imputation action has one of three search types associated with it. Two of these search types, binary and partition searches, are used to perform imputation actions for which only 1 matching variable is specified. The other search type, the multivariable search, is performed when 2 or more continuous matching variables are to be used.

4.2.1 Type 1 — Partition Searches

Partition Searches are performed when only 1 discrete matching variable with a small number of possible values is specified for the imputation action, e.g., as in the case where a respondent reports the total number of tractors, but neglects to give the corresponding total dollar value. Since a farmer is unlikely to have more than 3 tractors the donor population is divided into 3 partitions: 1, 2, or 3+ tractors. A donor is chosen at random from the partition to which the recipient record belongs. If there are no donor records within the partition to which the recipient record belongs, but there are donors in any of the subsequent (higher numbered) partitions, then all of the subsequent partitions are collapsed into one and a donor record is selected at random from this collapsed partition. If there are no donor records in the partition to which the recipient record belongs or in any subsequent partition, then a donor record is selected at random from the closest preceding (lower numbered) partition that contains any donor records. As these collapsing procedures are not frequently applied, no serious introduction of bias is encountered. If the donor population is empty, then the field to be imputed is assigned the maximum value allowable by the edits and the record flagged to indicate that imputation was unsuccessful. These flagged records are then reviewed by subject matter personnel who manually assign an appropriate value to the field requiring imputation.

4.2.2 Type 2 — Binary Searches

Binary searches are performed when only 1 continuous matching variable is specified for the imputation action, e.g., as in the case where a respondent reports the total value of his/her tractors, but does not give the corresponding number of machines. The entire file of eligible

donor records is searched and the record that minimizes the difference between the matching variable values is selected as the donor. If two or more potential donor records are equally close, then the one that is geographically closer to the recipient (as judged from the geographic ID) is automatically selected as the donor. If the donor population is empty, then the recipient record is flagged to indicate that imputation was unsuccessful.

4.2.3 Type 3 — Multivariable Searches

Multivariable searches are performed when more than one continuous matching variable are specified for the imputation action. These are the most complex of the three search types performed by the 1981 Census of Agriculture. The method used to perform multivariable searches was adapted for use at Statistics Canada by G. Sande.

When the missing data are related to more than one continuous matching variable, it is desirable to use as a donor a record that is closest to the recipient record on all these matching variables simultaneously. This requires a multivariable search on a large donor file and has been made practical by grouping the donor population in such a way that it is not necessary to search every donor to determine the closest. This specialized grouping of records is called the K-D (Key Discriminator) tree. The same K-D tree may be used for all records requiring a certain donor IA within a particular imputation region as the file of eligible donors will remain the same in each case. However, if a different donor IA is to be performed using a different donor population, or even the same donor IA on a different imputation region, a new K-D tree must be built as the file of eligible donors will not contain the same records.

a) Building the K-D Tree

The first step in the building of the K-D tree is to perform a transformation on all of the matching variables by subtracting the mean and dividing by the standard deviation of the donor population. This allows matching variables of different scales to be specified for the same search.

After the variable transformation, the following algorithm is then used to actually build the K-D tree. It is first applied to the entire file of eligible donors, and then to all subfiles subsequently created by the algorithm.

Firstly, the range (largest value minus smallest value) is calculated for each of the matching variables specified. The median value of the variable with the largest range (or the variable with the smallest ID if there are 2 or more with the maximum range) is then calculated. The variable for which the median is calculated is called the discriminator variable. This median value is used to split the file into 2 new subfiles, the left subfile containing records with values less than or equal to the median value of the discriminator variable, and the right subfile containing records with values greater than the median value of the discriminator variable. The algorithm is then progressively re-applied to the resulting subfiles using all specified matching variables until all files become TERMINAL, at which point the building of the K-D tree is complete. A subfile becomes TERMINAL when either the range equals zero for all matching variables, i.e., all records in the subfile are identical, or if there are 16 or less records in the subfile.

The above algorithm will yield a K-D tree of the form illustrated in Figure 4.

Every record contained in the original file will be present in one and only one of the subfiles corresponding to the terminal nodes.

b) Searching the K-D Tree

In order to locate the best possible donor, it is necessary to decide which of the terminal nodes "corresponds" to the recipient record. This is done by traversing the K-D tree, using the transformed matching variable values of the recipient record, starting with the root node and proceeding until one of the terminal nodes is reached. At each node of the tree it is determined, using the discriminator variable for that node, which of the two lower nodes the recipient record corresponds to. The K-D tree is traversed in this manner until a terminal node is reached.

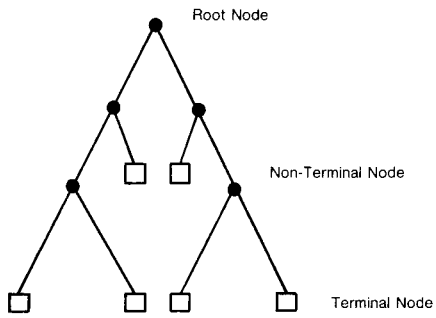

Figure 4. General Form of K-D Tree

In order to determine which donor in the chosen terminal node is closest to the recipient record, a distance function is required. Because of its ease of implementation, the distance defined by the maximum of the absolute differences between matching variables was used. The selected donor record is the one that minimizes this "distance".

Although the selected donor record is the closest to the recipient record contained in the chosen terminal node, it is possible that there are closer donor records residing in other

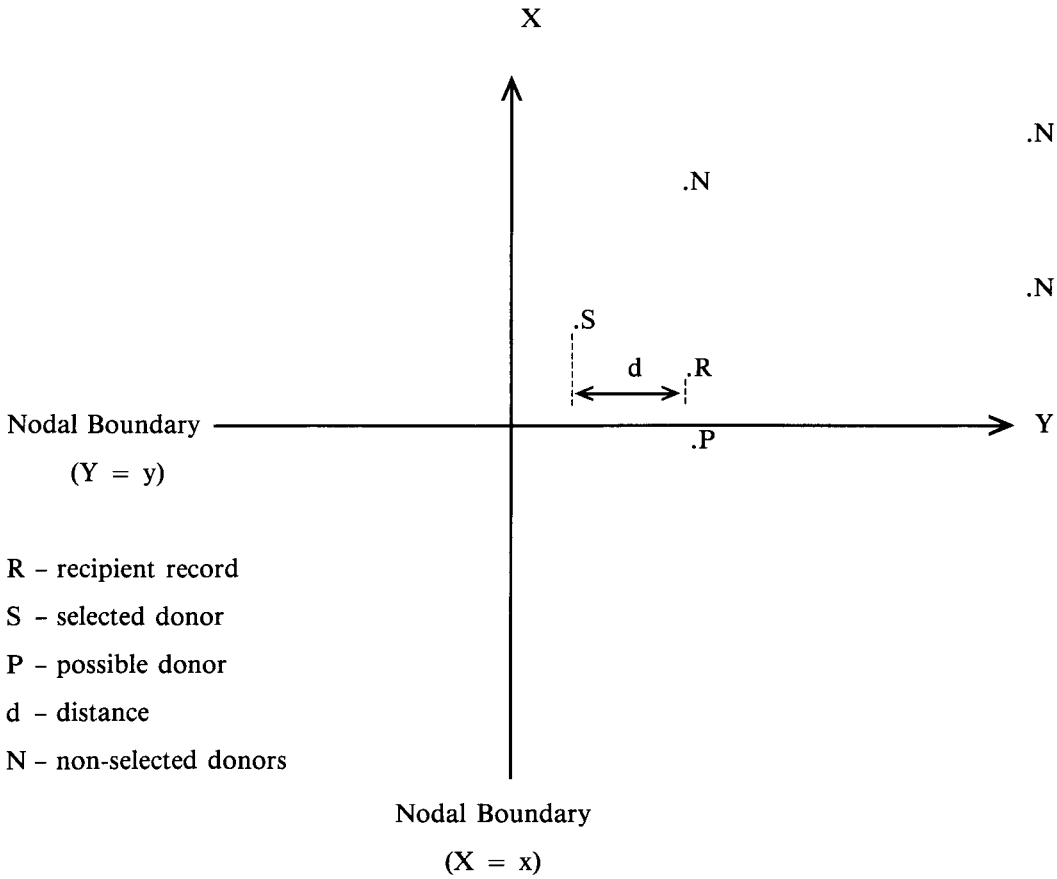

Figure 5. Closer Donors From Other Terminal Nodes (two matching variables)

terminal nodes. This may occur only if a nodal boundary exists that is closer to the recipient record than the currently selected donor record. This case is shown in Figure 5 for a donor IA involving two matching variables; X and Y. Each quadrant represents a terminal node.

It is evident that the possible donor P is closer to the recipient R than the selected donor S. This is possible because R is closer to the position of the nodal boundary Y = y than to S, and only donor records lying in the same terminal node as the recipient record may be selected.

A procedure, based on the variable values used to define the nodal boundaries and known as the bounds-overlap-ball (B.O.B.) test, is used to determine which of the other terminal nodes, if any, may contain donors closer to the recipient record than the selected donor record. Only terminal nodes that have the potential to provide closer donors are tested, and if a closer donor is found, then it replaces the previously selected donor. The B.O.B. test is applied until all nodes that may contain closer donors have been tested.

Finally, for all three search types, after the eventual donor record has been selected, the donated data values are prorated as previously described, if specified in the IA control information.

It will always be possible to select a donor unless the donor population is empty. If this occurs then the imputation region is collapsed with another and imputation is redone. It was never necessary to perform this operation in 1981.

5. CONCLUDING NOTE

A detailed evaluation, Grenier (1983), indicated that a major portion of the edit system was of little data quality benefit. This was because the Correction of Rejects procedures were unable to correct a sufficient proportion of the edit failures. For example, Correction of Rejects was unable to correct the failures resulting from a subset of 77 of the 97 edits more than 5% of the time. Also, many of the edits affected less than .1% of the population. Additionally, the Correction of Rejects procedures were highly labour intensive and created a heavy paper burden. To eliminate these inefficiencies a new computer edit system will be designed for 1986.

Statistics from the 1981 Census of Agriculture, Grenier (1983), indicated that 43% of the farms in Canada had at least one field imputed. Of this 43%:

18% required internal imputation only,
17% required donor imputation only, and
8% required both internal and donor imputation.

An analysis of the data distributions before and after imputation indicated that the imputation system did not have a serious impact at the Canada level although many of the 137,390 records imputed underwent a significant change. The system successfully handled all necessary imputations with only 58 records requiring manual imputation. The system was found to be very efficient, a processing cost of only $15,000 being incurred. Diagnostic data indicated that minor modifications to the system must be made for greenhouses, mushroom houses, community pastures, and institutions, if they are to remain in the census. Due to its successful fulfillment of the requirements, it is planned to reuse the present imputation system in 1986.

REFERENCES

SHIELDS, M., and YIPTONG, J. (1981). Census of Agriculture-1981 Imputation Specifications. Technical Report, Census and Household Survey Methods Division, Statistics Canada.

GRENIER, A.R. (1983). 1981 Census of Agriculture Evaluation Report. Technical Report, Agriculture Statistics Division, Statistics Canada.