Searching for Donors: Refining an Imputation Strategy

Michael Hogye and Peter Quan
National Agricultural Statistics Service, USDA, 3251 Old Lee Hwy, Fairfax, VA 22030

Abstract
For editing data from the 2007 U.S. Census of Agriculture, USDA/NASS built upon the foundation it laid for the 2002 Ag Census, when it ventured for the first time into editing and imputation on a large scale. Institutional acceptance of automated data correction was hindered in 2002 by problems inherent in establishing an entirely new enterprise-level processing system. Changes for 2007 were thus focused on making the system more responsive and reliable. This paper is an overview of improvements facilitated by development of a new imputation subsystem. New tools were provided---within the framework of decision logic tables (DLTs) built by NASS---not only to make donor data available directly to DLTs, but also to improve donor selection, especially through stratification. A variety of SAS techniques achieved better speed in identifying and retrieving nearest neighbors.

Key Words: imputation, donor, donor pool, editing

1. Introduction

The National Agricultural Statistics Service (NASS) of the U.S. Department of Agriculture built a new edit and imputation system to process data from the 2002 Census of Agriculture. During the 1997-2002 preparation cycle, the editing project was a major step in completing the transition of responsibility for the Ag Census from the Bureau of the Census to NASS (Yost et al, 2000). Birth pangs during the 2002 cycle evolved into growing pains that punctuated the 2007 cycle, whose improvements have been documented elsewhere (Atkinson and Beranek, 2008). This paper focuses on how the system's donor imputation function was redesigned between 2002 and 2007.

In keeping with the NASS motto of producing "timely, accurate and useful statistics in service to U.S. agriculture," staff members in offices throughout the country have used their subject-matter expertise to edit surveys for over a century. The task of editing Census data has presented special challenges to uphold that standard. Development and implementation of an editing system for the Census has been a unique experience for NASS. To be timely, editing and imputation must be automated as much as possible. This is necessary because the Census is an order of magnitude larger than traditional NASS surveys, producing approximately 1.5 million final records. To be accurate, hundreds of data fields in each report must be examined for internal consistency and reasonableness. For the automated process to achieve this, edit logic must be formally codified to make necessary corrections and perform imputations. To be useful, aggregated data must be evaluated for outliers and compared against historical baselines or other data sources. For this purpose, uniform tools for assessing data quality must be made available to analysts nationwide.

Preparations for the Census added new dimensions to the NASS culture. Until the advent of the Census, automation had generally been used only to flag erroneous data and to leverage individual subject-matter expertise for manual correction rather than to directly apply changes to the data. Although strata averages are sometimes automatically

computed and used for imputation, this is generally a fallback strategy. As a facilitator in the interaction between NASS edit staff and raw survey data, the primary role of automation has been to identify problems so that analysts can make corrections. To meet Census targets for timeliness, much of this authority to alter data must be ceded to an automated process. To meet NASS standards for accuracy, the automated changes must establish their credibility in the eyes of NASS staff. Finally, usefulness of the Census data ultimately hinges on additional automation which highlights troublesome records, and then provides an interactive opportunity for NASS staff to use their experience and NASS databases in correcting those records.

The editing and imputation system which NASS put in place for processing the 2002 Census got the job done: data were successfully edited and deadlines were met. Nevertheless, NASS staff nationwide spent long hours compensating for system shortcomings and endured many frustrations overcoming system limitations. First, circumstances conspired to force much of the testing and production to proceed simultaneously. Then, the proportion of records which required personal attention by analysts was much higher than the goal which the agency had set. The only remedy was for analysts to personally shepherd such records through the system using a fast, highly reliable interactive edit capability. Although such a tool was created to provide a high-quality interactive platform for editing, the requisite speed and reliability were lacking. Following the 2002 Census, NASS concluded that "Employees must be provided a significantly improved system for the 2007 Census with respect to the speed, stability, and quality of the data generated by the interactive edit/imputation system" (Nealon, 2004). While a major step had been taken to establish NASS' first truly automated edit and imputation system, the 2002 edition of the system fell short in gaining the full trust of NASS staff. The role of imputation in this disappointment was substantial, setting the stage for changes in imputation for 2007.

2. Overview of Editing in the 2002 Census of Agriculture

The 2002 Census edit processing was centered around multiple instances of a batch SAS program running on a large AIX system. Each copy of the program had the capacity to edit seventy-five Census reports from the same state. This software was named the “wrapper,” because it not only supervised the editing of each record, but also used SAS procedures to wrap required resources around each module of the edit code. The wrapper would begin by collecting data values from a RedBrick database, merging them with administrative data which it pulled from a separate Sybase database. Under control of the wrapper, edit logic was applied to the data values through a sequence of 46 modules. Each module evaluated a portion of each data record, while accumulating data updates, warnings and administrative information into temporary SAS datasets. The wrapper would conclude by posting error codes, status flags, tracking data and other system updates as transactions to Sybase. It would also create transactions for updating the data values themselves; however, these changes were placed in a queue and then applied to the RedBrick database by a special loader program.

The wrapper program was also a conduit to other edit resources. For some farm operations, previously reported data (PRD) were available in a separate Redbrick database, to be used at the discretion of the edit program in place of missing or incorrect values. If PRD were unavailable or inappropriate, then---in more than half of the edit modules---a donor imputation program could be invoked to request data from a similar farm whose 2002 Census data were already available. A SAS macro named Nearest

Neighbor Imputation (NNI) was written for this purpose and incorporated into the wrapper.

Limitations to the 2002 edit system design became evident as processing progressed. Edit processing in general and analysts' edit work in particular were hampered by slow system throughput. After a mandatory run through the automated edit, each census record became available for personal examination and correction by NASS analysts using an interactive SAS program devised for this purpose. Known as Data Review, this facility tapped into both the Sybase and Redbrick databases, displaying data for a specified farm operation and offering the analyst an opportunity to make corrections. But changes had to be posted to the databases and then run through the wrapper's edit process again as a batch job. Delays in these steps prevented analysts from following up on individual records in a timely manner. Changes passed to the RedBrick loader from both the initial batch edit and Data Review often developed a backlog. Confusion arose when newer Sybase administrative data for a record were not consistent with older RedBrick data values which were waiting to be updated.

Each wrapper process was complex and resource intensive. SAS Component Language (SCL) to process each edit module alternated with the wrapper's base SAS instructions, saddling an innovative edit design with inherently slow performance. Although written for maximum efficiency, the NNI program was nevertheless a SAS routine whose processing time added significantly to the length of edit jobs where it was called upon to provide imputation. Amplifying these limitations in editing speed was the need to run multiple batch edit jobs concurrently, generally causing contention for system resources. On average, each record took over a minute to run through the wrapper. Even this time was short, however, in comparison to the typical time it took for a wrapper process to advance to the head of the job queue, as well as the additional time lag for loading updated data values into the RedBrick database. It was difficult for an analyst to anticipate whether interactive changes made to a record would be batch-edited and posted for reexamination with Data Review within an hour, a day or a month.

The wrapper program was just one of many new tools that NASS developed for editing the Census of Agriculture. Subject-matter experts were trained to write decision-logic tables (DLTs) for each edit module. A suite of applications to create, test and run DLTs was developed by Robert White of NASS (White, 2003). An explicit syntax was constructed by the developer and then used by specialists to enter DLTs into the system with an Authoring Tool program. After screening and validation by the Authoring Tool, the DLTs were compiled into a SAS SCL list. The Evaluator and Runner were tools developed to read the stored DLT instructions from the SCL list, interpret them, and carry them out. By supplementing the DLTs with a final opportunity to correct data, the NNI program became the first NASS venture into automated donor imputation. As a standard tool for NASS staff nationwide, the new Data Review facility allowed individual analysts in remote locations to pool their expertise to correct data. These and other initiatives allowed NASS to complete an editing effort unparalleled in NASS history (Atkinson, 2003). In spite of their leading-edge credentials, however, these resources left most users with one overriding impression: they were too slow.

To improve the edit system's performance, three major changes were contemplated for editing the 2007 Census. First, editing transactions would be handled entirely by a single Sybase database containing both administrative and edited data values. All updates to the same Census record would be loaded simultaneously to one destination, not only

precluding inconsistencies between databases but also eliminating delays caused by the RedBrick loading process. As a transactional system, Sybase was considered better suited for making large volumes of changes. As an analytical system, RedBrick was to be used for additional review of edited data, but not for rapid posting of small changes. A second desired change was to make Data Review truly interactive. Analysts would be able to immediately run the edit program on their local computers as soon as they made changes, rather than waiting for database updates and remote batch edit processing. Before posting changes to the database, analysts would be able complete their review of a record, making all changes in one session without interruption. A third major goal was to speed up the edit program itself by at least an order of magnitude. A faster wrapper program would result in greater throughput for normal batch editing, while improved edit performance would also greatly benefit local processing during Data Review. Faster batch processing would allow better management of system resources, with flexibility not only for scheduling system maintenance and upgrades, but also for recovering from unplanned down time. Less pressure for round-the-clock, full-capacity processing would encourage smoother overall edit processing.

The NNI program played a role in areas affected by all three major proposals. First, the donors themselves were being pulled from the databases, so that restructuring of the Sybase and RedBrick environments would affect the way donor pools were assembled. Second, by being upgraded to include the full edit, Data Review would require donor imputation capability within its locally-run edit processes. Third, as a major component of the overall edit process, donor imputation processing would have to be significantly speeded up in order to meet the proposed targets for edit speeds. As an additional goal, it was hoped that overall data quality for records processed by the automated edit would improve. This was especially desirable in order to reduce the burden on analysts to make manual corrections, which in turn would add momentum to the effort to speed up processing. Since many of the questionable data values inserted by the wrapper program were obtained from NNI, it was also hoped that there would be improvements in the imputed values themselves.

3. Imputation in the 2002 Census of Agriculture

A major goal in building the 2002 editing system was to automate as much as possible the process of correcting records---making them internally consistent by forcing them to obey explicit edit rules. The wrapper applied edit logic in increments by looping through a series of modules and passing control to the Runner and Evaluator programs to deal with each module. In each cycle of the loop, the stored DLT code relevant to the module was retrieved and carried out, executing steps that had been laid out in DLT format. For NASS, not only were DLTs a new approach to editing, but they also broke with precedent because they were given authority to change data values, even before analysts had an opportunity to see them. Throughout its processing, each wrapper program maintained a “current value” table, as well as a table of PRD available for those records. As each module was processed, the DLT logic first identified problems and then determined how errors might be corrected. There were three options for making changes. The DLT could specify new data values that appeared reasonable based on context; it could substitute values based on PRD; or it could flag fields needing corrections and defer to the NNI program to make changes based on data from a similar farm.

A special Impute function was designed for the DLT to request an imputed value. Its syntax is of the form

IMPUTE(*target-var, positive-flag, ratio-var*) where:
target-var is the field to be imputed,
positive-flag notes whether a positive target value is required, and
ratio-var provides an optional field for adjusting the donor value before imputing.

During the editing of a Census record, Impute calls within an edit module collectively specify the parameters for a single donor search. For each search, all of the imputed values are taken from the selected donor. (In 2002, there was only one search possible for each module.) The job of the DLT is to trigger the imputation process by accumulating the arguments from Impute requests. In 2002, the wrapper would place -1 into the current value table wherever imputation was anticipated, and send a table of imputation instructions to the NNI program. NNI would then honor the requests by giving the wrapper a table of updates for the imputed fields. The wrapper would replace each -1 with an imputed value whenever a suitable donor was found. Thus as the 2002 edit proceeded, control would cycle from DLT execution under the Runner and Evaluator; to imputation under NNI; to current-value updates under the wrapper; to DLT execution for the next module.

In 2002, donor data were organized into SAS datasets, which the NNI program searched for a nearest neighbor. Each dataset contained successfully edited Census records for a state, including records from adjoining localities in adjacent states. Each dataset grew as more records were cleaned and as there were opportunities to extract them from the databases and add them to the donor datasets. When a record required imputation, the dataset of donors for its state was searched for a nearest neighbor, taking account of the restrictions imposed by the Impute calls. The data organization and search techniques used for 2002 have been documented elsewhere (Hogye, 2004).

When NNI was called by the wrapper at the conclusion of an edit module, a donor search was conducted for each record requiring imputation. Each search in turn was carried out using the matching variables and positivity restrictions collectively implied by its imputation requests. Distances between eligible donors and the recipient were computed as the sum of the squared differences in the values of their normalized matching variables. Before moving on to the next edit module, the wrapper would adopt imputed values from NNI into its current-value table, but only if the imputed record conformed to a group of "relaxed linear edits." Thus it was the responsibility of NNI to provide imputed values which satisfied these minimal edit constraints. To improve the chance of meeting this requirement, NNI would identify up to 25 nearest neighbors during its donor search; substitute their values into the recipient record; test the record against the relaxed edits; and send to the wrapper updates based on the nearest neighbor that passed the test. In the event that all available donors failed either the positivity constraints or the relaxed edits, NNI would signal the wrapper that donor imputation had failed. In such cases the wrapper would halt the processing of the Census record and set a flag in the database indicating that analyst intervention was required.

4. Considerations for Imputation in the 2007 Census of Agriculture

NASS' experience with imputation in the 2002 Census influenced preparations for editing the 2007 Census. The idea of building a pool of clean records to provide donors from the ongoing Census was expected to be used again. The principle of using the same donor to impute multiple values in the same record was also considered desirable. But

there was considerable dissatisfaction with imputation by the NNI program, aggravated by the inconvenience of what it took to manually override it. Many of these deficiencies were generally summarized as a need for better data quality in imputation. Of greatest importance, however, was the unanimous feeling that the edit as a whole and imputation in particular had to run much faster.

Soon after processing of the 2002 Census was complete, changes to the edit infrastructure provided immediate benefits for 2007. A simpler data model was made possible by directing all of the edit's transactions to a single Sybase database. After each Census data record was introduced into the database, it was subjected to a mandatory initial edit, whose results were all posted as updates to the same database. If any further edits were applied to the record, then only additional Sybase updates were applied. Although it put great demands on the single database, this consolidation eliminated problems with lag time and synchronization between databases, making a record available for additional editing as soon as a previous edit was concluded.

In another change, the edit process itself was made more efficient by restructuring the wrapper. In the original program, unnecessary overhead had been introduced by repeatedly detouring the wrapper from base SAS into SAS SCL for each module, using Proc Display to run DLTs. To avoid restarting the SCL mode of processing in each module, much of the wrapper itself was rewritten as an SCL program. Under the new framework, once the principal wrapper program began to run in SCL mode, the Runner and Evaluator SCL programs carried out each module's DLT in turn, without exiting from SCL mode. Although the SCL often executed base SAS code through the use of "Submit Blocks," it proved to be more effective when it was allowed to keep overall control of the edit process. A related change was the decision to process each Census record completely before beginning the next record, rather than processing a batch of records simultaneously within each edit module. This not only simplified the flow of data, but also made it easier to track the progress of an individual record through the edit.

In spite of initial improvements, an intensive review of the edit software concluded that inherent design limitations were likely to keep editing speed well short of goals for the 2007 Census. Consultants' recommendations focused on alternatives for reformulating the way DLT instructions are carried out. These ranged from a different algorithm for interpreting DLT code, to a SAS code generator rather than a DLT code interpreter, to compiled programs written in other languages and integrated into the SAS-based wrapper. Streamlining the edit was intended to not only bolster the performance of large-scale batch processes on the AIX system, but especially to allow PCs to run the edit as well. For the 2007 processing environment to provide a truly interactive editing capability, the wrapper, DLTs and imputation would all have to be available on demand to NASS analysts around the country. However, even as the commitment was made to "port" the edit to the desktop, indications were that its performance could not meet expectations.

Compared to the scrutiny given to DLT processing, few suggestions were offered for speeding up imputation. As a base SAS application, NNI did not invoke SCL and did not require the code interpreter used for DLTs. Nevertheless it consumed a large portion of the overall edit processing time. Since imputation was called upon very frequently, often several times within the same record, it could easily account for the majority of a record's editing time. Although it was programmed to minimize the time to carry out individual steps, NNI was still forced to run many SAS data steps and SAS SQL procedures each

time it was called. The program had to evaluate imputation parameters; derive lists of matching variables and positive constraints; formulate a donor search; do the search while tracking 25 nearest neighbors; create 25 copies of the recipient to contain substituted values; apply as many as several dozen linear edits to all 25 copies; and create a file of final imputed results. The time taken by donor searches could be reduced by limiting the size of large states' donor pools. This was done during the 2002 processing, but it only prevented unreasonably long imputation times from becoming totally unacceptable.

Planning for 2007 imputation was complicated by plans to make Data Review fully functional by adding the option of locally running a truly interactive edit. For 2002, nearly 100 large SAS datasets had been created and regularly updated on the main AIX system to provide donors. For 2007, would each local system or each state office maintain its own dataset of donors? Would it include all donors for the whole state? If not, which ones should it include? If local donor pools were used, how would they be created and updated? If a single, central donor pool on the AIX system would be kept for 2007, then how could it be searched even faster from a remote location than when the search process was on the same system?

In spite of the obstacles pointed out in system evaluations, NASS staff made steady improvements to the speed of DLT processing. At an early point in the preparation cycle for the 2007 Census, the processing speed for editing a record came within striking distance of the target. However, these test times did not include any imputation, because the new version of the edit did not yet contain any software to handle it.

5. Preparations for Imputation in the 2007 Census of Agriculture

A starting point for 2007 imputation planning was the management of data used for imputation, a subject referred to here as *donor maintenance*. In order to avoid interfering with critical Sybase operations, a separate database was established to hold all data eligible for use as donors. This central donor repository (CDR) was a RedBrick database designed to allow full investigation of imputation matters by segregating it from other uses. At first it was populated with final data from the 2002 Census. Simply having a source of donors during development was a major advantage over preparations for 2002, when no such information was available either for testing or for use as a starting donor pool. Once processing of the 2007 Census began with the new data base scheme, successfully edited "clean" records were automatically copied from Sybase and set aside in Redbrick for use with analytical applications, with minimal disruption to editing operations. As one of these applications, the CDR concept fit in well. It ultimately evolved into a copy of all 2007 data eligible for imputation use. Before production, when records were received and edited during the formal Content Test of the 2007 Census, clean test records were added to the CDR. In addition, many records from the 2002 Census were modified to resemble 2007 Census records for use in testing the system. They were re-edited with the new 2007 system and then added to the CDR. The re-editing made use of Content Test records as donors whenever possible, with original 2002 data as a fallback. This substantial test of the new editing system also helped to create a starting donor pool for the 2007 Census. As a clearing house for donor data, the CDR handled general donor maintenance issues, while the testing process allowed it to evolve as more clean records became available.

The role of imputation in the edit/wrapper program was changed. It was agreed that

imputation should be carried out as a DLT action whose results could be reviewed by other DLT rules in the same module. This meant that imputation would happen under DLT control, rather than being an independent step following completion of each module's DLTs. For this purpose, there would be a "Get Donor" action to request a donor and impute its values into the fields targeted by earlier Impute calls. Once a Get Donor call was executed, the current-value table would be updated but still subject to further changes dictated by DLT logic. In conjunction with other DLT changes, Get Donor was added to the list of actions recognized by the Runner and Evaluator programs. The more detailed question of how Get Donor would actually function to get a donor was deferred.

The methodology for determining nearest-neighbor distances was also changed. Rather than combining groups of matching variables tied to each of the target variables, an explicit list of modest length was provided by subject-matter experts for each donor search. As DLT authors wrote Impute calls, they now had the opportunity to specify a fourth argument, dictating which matching variables to use when a subsequent Get Donor search was executed. As in 2002, distances were derived as the sum of squared differences, but the use of fewer matching variables made them simpler and more intuitively reasonable. A particularly useful case was when a ratio variable was used to scale multiple imputed values to a specified sum. Under the new system, the ratio variable could also be used as a primary matching variable, steering the donor search toward a record with a similar total and thus minimizing the distortion caused by scaling the component values borrowed from the donor.

Special matching variables were also included. Geographic location influenced 2002 Census donor selection through the inclusion of latitude and longitude among many matching variables. For the 2007 Census, an approximate recipient-to-donor mileage was computed and used as a matching variable. Its contribution to the total matching-variable sum of squares was weighted to give it an influence very roughly equivalent to that of the other matching variables combined. Other variables were made available to weight donors according to their age, usage and imputed content. Adding a penalty to older data proved useful, for example, in favoring the selection of new, 2007 Census records over donors from earlier sources. The number of times a donor had been previously used for imputation, and the percentage of the donor which was itself imputed, were both tested as additional quantities to influence donor selection. Difficulties in calibrating these effects combined with other practical issues to defer their use until they could be better refined in future research.

6. Stratification of Donors in the 2007 Census of Agriculture

In the 2002 Census edit, all donors in a recipient's state---plus some from adjoining states---were considered for imputation. Selection was restricted by positivity constraints, which effectively limited donor selection to implicitly defined strata based on the presence or absence of certain quantities. But for each request, all records of a state-wide donor pool had to be considered and there were no explicit strata within the state. After the 2002 edit experience, this practice was challenged on the grounds that it was time-consuming and that it was susceptible to inappropriate donor selections. There was strong sentiment for reducing the scope of records examined by each donor search, with the hope that imputation could focus on appropriate donors without individually considering all those available.

This issue led to questions which balanced theory against practice. One of the Fellegi-

Holt principles is the ideal of preserving true data distributions as corrections are made to individual records (Fellegi and Holt, 1976). In this regard the Census of Agriculture faces serious obstacles due to its broad range of content, its complex data distributions, and its extended editing timetable. At the start of editing for 2002, nothing was known regarding data distributions; there simply were no data available. At the start of 2007 editing, data were available from other sources, but there were no current data. As editing progressed in each Census, the base of current information expanded in step with the receipt of Census forms; the order of their processing; and the rate of success in correcting their data. Over the course of several months' processing, donors were added to the imputation repository as they became available. It was only as the donor pools matured that they began to paint a reliable picture of how current data were distributed. Nevertheless imputation was required from the beginning of editing, so that early imputations had to be based on the best data available, while later imputations could benefit from a larger pool that was more representative of true distributions.

A related question that persists throughout the editing is how to choose an appropriate donor, faithful to the underlying distributions, regardless of the donor pool's condition. Among the wide variety of imputation strategies enumerated by Sande (Sande, 1982) are two which operate from the perspective of using donor data from the current survey. When the population falls into demographic categories such as gender and ethnicity, it is natural to select a donor from the recipient's group. Sande points out that a single donor (for all imputations in the same record) may be randomly chosen from the group, or pseudorandom choices may be made---one variable at a time---from different donors. He distinguishes the choice of a nearest neighbor based on continuous variables from this "hot deck" approach, but points out that it is a logical extension of the same concept. The common purpose of such tactics is to pick donor data likely to be harmonious with the population distributions as represented by the donor pool. Known information about a portion of the record is used to choose a donor or donors that will best fill gaps in other portions. The hot deck builds on the assumption of at least modest homogeneity within categories, for those fields targeted for imputation. Instead of identifying an appropriate category containing similar donors, the nearest-neighbor search identifies similar donors by use of a distance measure based on continuous data. Randomness draws from homogeneity within a similar category, while distance computations follow a path based on similarity.

The dilemma for 2007 was to find a balance between these. Farms are often classified by NASS as one of 16 standard Farm Types, and are also often characterized in terms of their total acreage (K46) as well as their total value of products sold (TVP). Within each edit module some combination of these three characteristics is considered likely to influence the imputed variables. Initial research efforts made use of all three factors to assign donors into strata with roughly equal membership. However, to achieve a specified level of estimation precision, NASS historically has stratified survey data into homogeneous subpopulations rather than into equally-sized groups. In a similar spirit, an algorithm was created for Census donor data, to group farms of similar type, acreage and value as considered appropriate for each edit module. During editing, records needing imputation were classified into these strata, which were searched for appropriate nearest neighbors.

For the 2007 Census it proved more practical to create a donor dataset for each module rather than for each state. Only records and data fields relevant to the module were included. For example, only operations with cattle were kept as potential cattle donors,

and for those donors only variables required for cattle imputation were part of the dataset. For each module, the donors were stratified within state. According to their relevance and availability within the module, the variables Farm Type, K46 and TVP were used in some combination to further stratify records within each state. For several modules that came early in the editing process, the K46 acreage alone was used. For some later modules, after an intermediate module established a reliable TVP value, Farm Type was used together with TVP or K46. The collection of records for each module and state was broken into subgroups, one step at a time. When Farm Type came first among the stratification variables, then its categories formed the initial strata within state. When applicable, several steps employing widely endorsed rules of thumb were used to create subgroups based on K46 and TVP.

A first step in creating subgroups from continuous variables was to identify any natural gaps in terms of the K46 or TVP values. While this general principle is widely advocated (Deming, 1960) the specific rule advanced by Vogel (Vogel, 1986) was used. A gap of more than two standard deviations between consecutive values became an interval acting as an empty stratum between two that were populated. As a second step, each unbroken interval of values was broken into quintiles, using cumulative aggregation of either K46 or TVP values. Each of these new subgroups accounted for roughly one fifth of the total TVP or K46 in the original group, while any gaps between values from one quintile to the next were left open. After marking any natural gaps and imposing 20-percent subgroups, the stratification process considered whether the subgroups might be reorganized more compactly. In general, group members falling more than three standard deviations above their group mean were reclassified into the next higher group. In particular, the lowest-valued group was allowed to yield members to the next highest group, while its remaining values were iteratively reconsidered in the same fashion until membership in the group stabilized, with no values falling beyond the recomputed upper limit. Then the group with the second-lowest values would be considered in the same manner, potentially giving up members to the third-highest group. This process would move through all the groups, until the highest-valued group was allowed to hand its outliers to a special group, newly formed to hold them.

The stratification algorithm included some special considerations. Even groups of only two or three values could lose values to the next-higher group, if those values were more than double the next-lowest value. On the other hand, all the groups based on K46 or TVP were also allowed an opportunity to be consolidated. If t-tests of adjacent groups fell below a threshold, then the groups were joined together. A final consideration was the possibility that, even after all these stratification rules had been applied, some strata might still be unreasonably large due to the skewed distribution of K46 or TVP in many states. In fact the outcome of the process was unsurprising, with a few relatively large strata as well as many strata composed of just a few outliers. A scheme was devised for subdividing the large strata into substrata in case they proved excessively time-consuming to search. Additional variables besides Farm Type, K46 and TVP were used to sort donors within the large strata, so that the records were organized into a continuum of characteristics relevant to the module. With systematic sampling, subgroups representative of the edit module's content for the whole stratum were formed. Although these substrata were not required for the 2007 processing, the capability for creating them was available.

The stratification worked in partnership with constraints and matching variables to determine donor selection within a state. When a donor was needed, some combination of

state, Farm Type, K46 and TVP was used to classify the recipient into a stratum. (To distinguish this narrow use of stratification from its use in other NASS applications, imputation strata are generally called "profiles.") Donor records in the recipient's profile could be readily identified, and restrictions were applied to exclude records which did not have required positive values in the specified target variables and ratio variables. Further narrowing of donor selection was made possible for 2007 with the inclusion of a "special constraint" option as part of the Get Donor instruction in the DLTs. For example, if only cattle donors with beef cattle were to be considered, then the instruction would read:
"Get Donor(K804>0) ." The remaining donors within the profile were all considered, to determine the nearest neighbor among them.

7. Implementation of Imputation in the 2007 Census of Agriculture

The 2002 NNI program became obsolete in the course of planning for 2007. Although significant improvements were made in the speed of DLT processing, those techniques could not be applied to the imputation code, so that the need for much faster imputation could not be met by the 2002 system. Even though stratification would reduce the scope of searches, experience in 2002 processing had shown that searching only a sample of a state's donors still took too long. In addition to its speed limitations, the NNI code was incompatible with changes in system design. Instead of trying to upgrade the 2002 imputation code, a new strategy was used. The job of specifying donor requirements and substituting donor values into the edit record was delegated to the edit wrapper program, while the job of validating the imputed values was left to the logic of the DLTs themselves. This allowed the time-consuming problem of identifying and retrieving an appropriate donor to be addressed as a separate task. Thus preparations for 2007 imputation became a series of interlocking responsibilities. The new *donor delivery* function joined donor maintenance in a partnership of specialized tasks to manage donors, while other imputation chores were left to the edit.

A donor delivery solution was gradually achieved through experimentation with a variety of SAS programming techniques. A dedicated SAS batch process was allocated to each module requiring imputation, running continuously on the central UNIX system for the sole purpose of providing donors. Each such SAS program acted as a donor-retrieval "daemon" because an endless loop kept the job active, answering requests as they came in from edit jobs needing donors. Each module-specific donor program held all relevant data in memory, including stratification and matching variables to identify an appropriate donor, as well as target and ratio variables for the edit to use in imputation. Extensive system memory was used within the SAS job to store data in temporary arrays, which offer two major advantages over normal SAS arrays: they can handle larger quantities of data and their contents can be retrieved more quickly. While the entire donor pool for each module could be stored in RAM, the life span of the arrays holding the data was limited to a single SAS Data Step. Thus the donor search itself had to be carried out without invoking other procedures or new data steps that would cause the program to lose its grip on the donor pool data.

Although no special search trees were constructed, donor information was nevertheless organized to make effective use of the temporary, memory-based arrays. Information for searching through the donor pool, such as matching variables, was housed in a two-dimensional "Keys" array. A "Profiles" array containing record counts for each profile made it possible to store donors of the same profile as a contiguous group within the Keys array, and then to easily locate and search a specific profile of donors. The Profiles

and Keys information together provided a rudimentary indexing capability that gave rapid access to the profile's search data, while the search data then allowed efficient application of constraints and matching-variable distance calculations. To determine a recipient's profile, special SAS formats were created to rapidly classify a record based on the values of its stratification variables. To aid in obtaining approximate recipient-to-donor mileages as part of the nearest-neighbor computation, SAS formats were also used to quickly map counties into the latitude-longitude values of their centroids. To supplement the data used for searches, a one-dimensional "BV" vector held data values for the more numerous target and ratio variables, using a storage compression scheme to minimize memory usage. Information was extracted from this area of memory only after the identity of a donor of interest had been pinned down by the search, when additional data values defining that donor were required for the imputation itself.

Some of the SAS variables within the Keys array were used as bit maps to efficiently convey binary information. For example, searches within a profile were first narrowed to donors with specified positive values. Each donor in the profile included SAS data values whose binary representation as a string of zeros and ones provided a map, with the ones corresponding to the positive-valued members in a reference list of variable names. By expressing the positivity requirements of a donor search as a bit mask flagging the required positive variables, it was possible to quickly determine whether each donor's bit map satisfied the constraints in the bit mask. This was done by comparing the donor against the requirements, using SAS bit-string functions. In similar fashion, bit maps were created and stored in the Keys array for each donor to show which special "Get Donor constraints" it satisfied, in reference to a master list of those constraints. During the execution of a search, each member of a profile was accepted or rejected, based on the way that its positive-value bit map aligned with the baseline bit mask of positive constraints, and also based on whether the bit representing a desired special constraint was turned on within its special-constraint bit map. A similar scheme was employed to facilitate the storage of ratio-variable and target-variable data in the BV vector. For each donor, two columns in the Keys array guided the process. One field gave the index showing where the donor's positive data values began within BV, while the other provided a bit map highlighting positive data values, in reference to a master list of data variables. This freed the BV vector to track only the non-zero data values, providing substantial economy in memory usage---especially in certain edit modules.

While donor delivery was streamlined by the use of bit maps, it fell to donor maintenance to set up relevant variable lists as well as the SAS bit-map variables. As DLTs were developed, there were no fixed lists of target variables, ratio variables, special constraints and matching variables to choose from. Instead, this information was added to the arguments of Impute and Get Donor actions, as DLT developers saw fit. Thus each time there were changes in the edit logic which touched on imputation, it was possible for the metadata lists to change. For example, as more Impute calls were added to a module's DLTs, additional target and ratio variables were likely to be cited. When donor information for a module was extracted from the CDR, the values of those fields had to be included in the data downloaded for use in the temporary arrays. The reference tables of variable lists also had to be expanded to include the new variables, which in turn changed the relative positions of other variables and thus affected the bit maps whose meaning depended on them. To manage all this properly, the first step in each donor maintenance cycle consisted of a detailed search of the DLT instructions themselves, extracting all Impute and Get Donor arguments. These were parsed by module, allowing the creation of standard lists containing target variables, ratio variables, matching

variables, special constraints and variables within special constraints. These lists in turn contributed to the creation of appropriate bit maps and to the identification of data fields to be downloaded from the CDR. The downloaded donor pool information was saved into a SAS dataset for each edit module, making it convenient for loading the Keys and BV arrays as the donor daemon was activated.

A crucial part of donor delivery was the communication of edit jobs with each edit module's donor daemon. A complicating factor was that donors were required for local edits in Data Review, as well as for batch edits on the central Unix system. When an analyst would locally edit an individual record, at least one module was likely to require a donor. The donor would either have to be chosen and relayed from a remote system holding a central donor pool, or the donor would have to come from a donor pool maintained in the local environment. The former approach would require exchanges not only between batch edits and donor daemons, but also between local edits and remote daemons. The latter approach would introduce major donor maintenance issues at the local level, since large volumes of donor data and related metadata would have to be regularly updated across systems nationwide---simultaneously. Both approaches required rapid data exchanges between SAS jobs, as well as a mechanism for queuing donor requests from multiple edit jobs addressing the same daemon.

After experimentation with piping data directly through memory, a file-based solution was adopted. Special files under the control of a SAS/Share server process were introduced to mediate inter-process communication between an edit job and a donor daemon running on the Unix system. For each edit module, a special Requests file acted as the gateway to its donor daemon. Somewhat like an office in-box with a tray for each state, the Requests file allotted one record to each state. To submit a donor request, an edit would write---into the appropriate record of the Requests file---the recipient information needed to define the donor search. These specifications included data to classify the recipient into a stratum; bit masks to further restrict donor selection; and values of matching variables. SAS/Share allowed many activities centered on each Requests file to proceed in an orderly manner. Requests from different states were handled concurrently; competition among multiple jobs from the same state was resolved; and polling of the Requests file was conducted continuously by the donor daemon to detect new requests.

A similarly constructed Responses file returned a donor in answer to each request. An edit job would submit a donor request to a record in the Requests file and then monitor the corresponding record in the Responses file until it detected that the daemon had provided a donor. The returned donor information would either include values of target and ratio variables, or values indicating that an appropriate donor could not be provided. Using flags in the Requests and Responses file to establish a set of crude handshake and disengagement protocols, exchanges between edits and donors proceeded smoothly, with record locking under the control of the SAS/Share process. Local edits using Data Review participated in this process by using SAS/Connect to upload donor requests to the remote Unix system. Under SAS/Connect, the local SAS process would pass the request to a remote SAS process, which then wrote it to the Requests file; entered the same queue as batch jobs waiting for donor services; listened for a response by monitoring the Responses file; received the donor; and then downloaded the information to the local edit for imputation.

Twenty-six daemons ran over several months to deliver donors for editing the 2007

Census of Agriculture. Each daemon held data in memory specific to one edit module, and interacted with edit jobs through its own pair of shared files which acted as "in" and "out" boxes. Millions of donor requests from both batch edits and local edits were smoothly processed. In the background, donor maintenance functions were exercised, initially at weekly intervals, to update the donor pool information. In this process, DLTs were parsed for imputation-related metadata; the CDR was tapped for clean records; strata were established; bit maps were computed for each donor; matching-variable variances were computed for each stratum; and datasets were created to house the donors. At opportune times, edit processing was briefly suspended while the donor daemons were halted and restarted with fresh data. Over the course of edit processing, the twenty-six donor pools rapidly evolved from "cold" data to a mixture that progressively included more and more current data, until their data were all exclusively from the current Census.

Imputation became a more effective tool under the control of DLTs. Subject-matter experts were able to program a wide variety of scenarios that included donor imputation as well as steps to override it when appropriate. The choice of donors was improved through deeper stratification, more flexible constraints and better use of matching variables. The utility of interactive data review was significantly improved by the addition of a streamlined local edit on the analyst desktop, including access to the same donor daemons that serviced the batch edits. The institution of standard donor maintenance procedures made it possible to plan regular cycles for refreshing donor pools. Because they were independent of the edit programs, these planned steps worked well. By preparing the data ahead of time, they made possible rapid donor delivery from memory-based donor pools to many edit jobs simultaneously. As a result, there were no backlogs in providing imputation or in meeting edit production targets. Imputation did its part in meeting the preeminent NASS goal of providing a faster and more reliable edit system.

8. Looking Ahead from 2007, to the 2012 Census of Agriculture

Having met agency expectations, the new infrastructure for donor maintenance and delivery will probably be used again for the 2012 Census of Agriculture. Preparations for the next Census should include investigations into the statistical properties of imputed data, leading to more robust methods for using what is known to impute what is unknown. Several options for deriving a donor from the donor pool were programmed for 2007 but not exercised. These included the application of Mahalanobis distance to the matching variables; random donor selection within profiles; and weighted random selection. An extension of these options allowed for the aggregation of multiple nearest neighbors into a consolidated donor using either means, weighted means, or an idea based on what is often called the Fermat-Weber solution (Keller, 2004). If a standard NASS methodology for evaluating imputed outcomes can be developed---perhaps using some variant of multiple-imputation techniques---then alternatives such as those already programmed may be assessed and considered for future use. At the same time, donor-pool data might be used to develop predictive models to provide imputation. Finally, once tools are in place to evaluate imputation techniques, a variety of stratification alternatives may be comprehensively tested for each edit module, either as part of a predictive model or as part of a donor-selection process.

The composition of the initial donor pools remains an open question, which NASS is in a good position to more thoroughly investigate after two Census cycles of experience. At the start of processing, there are effectively no current data to draw from. Even though

there is now a substantial accumulation of earlier data, it is generally not trusted to paint a representative picture of current data distributions. In addition, for Census information being collected for the first time it may not be possible to make any use of historical data for creating starting donor pools. If only current data are to be used for imputation, then considerable planning and development may be required to isolate useful donor data at the start of edit processing. A preliminary editing pass through the first batches of data from each state may be required, or a temporary fallback to hot-deck imputation during early processing may be helpful. An alternative may be to use early returns of current data to derive parameter estimates for adjusting distributions of cold data.

With the attainment of goals for speed and reliability, the platform which NASS built to edit Census data has matured significantly. That platform should now be available for use as a research tool, to create a standard set of procedures for assessing imputation, and ultimately to further refine the imputation methodology itself.

References

Atkinson, D. (2003), "*The Development and Implementation of a New Processing System for the 2002 Census of Agriculture*," UN/ECE Work Session on Statistical Data Editing, Madrid, October 20-22, 2003.

Atkinson, D. and Beranek, J. (2008), "*Further Improvements to an Edit and Imputation System for the 2007 United States Census of Agriculture*," UN/ECE Work Session on Statistical Data Editing, Vienna, April 21-23, 2008.

Deming, W. (1960), *Sample Design in Business Research*, Wiley and Sons, 487-490.

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

Hogye, M. (2004), "*Searching for Donors: Finding an Imputation Strategy*," JSM Proceedings, Toronto, August 8-12, 2004.

Keller, T. (2004), *"Imputation and the Fermat-Weber Problem,"* NASS Internal Memorandum, June 4, 2004.

Nealon, J. (2004), "*PRISM Interactive Edit/Imputation System*," Decision Memorandum for internal NASS use, November 2004.

Sande, I. (1982), "*Imputation in Surveys: Coping with Reality*," The American Statistician, Vol 36, No 3, Part 1, 145-152.

Vogel, F. (1986), "*Survey Design and Estimation for Agricultural Surveys*," SRS, USDA May 1986, 14-16.

White, R. (circa 2003), "*Prism System:DLT Edit System*," NASS Internal Documentation.

Yost, M., Atkinson, D., Miller, J., Parsons, J., Pense, R. and Swaim, N. (2000), "*Developing A State of the Art Editing, Imputation and Analysis System for the 2002 Agricultural Census and Beyond*," Processing Methodology Sub-Team Report, National Agricultural Statistics Service.