CONVERSION OF MULTIPLE SURVEY SYSTEMS' EDITS AND IMPUTATION TO StEPS

Paula Weir, Energy Information Administration, DOE
1000 Independence Ave. SW, Washington D.C. 20585, pweir@eia.doe.gov

KEY WORDS: Survey Processing System, Editing, Imputing, System Conversion

Introduction

The Energy Information Administration (EIA) operates over 100 different surveys that collect data on various types of energy at various points in the distribution flow from producers to users. These data address supply and demand issues by measuring production, imports, storage, sales, and consumption. Associated with these data are individual processing systems developed to accommodate the specific survey needs. These systems operate in multiple environments--mainframe, LAN and PC--and multiple languages and databases. While these surveys and systems have evolved over time, most are old and have been patched a number of times. Some of these systems have become very difficult to operate, causing problems with greater and greater frequency. It is clear that most need to be rebuilt. A more integrated approach to rebuilding the systems was considered for one fuel group of surveys. Resource limitations are the main factors driving the need for a generalized system. Even more preferable, if possible, is a generalized system that has already been developed, tested and is fully operational. As a result, EIA began considering the use of the Standard Economic Processing System (StEPS) developed by the US Census Bureau. This processing system includes modules for specifying parameters for the specific users and survey, modules for data collection activities including mailing, receipt and check-in, as well as modules for post collection such as editing and imputation. At this time, EIA is in the process of loading survey specifications and data for one of its surveys into StEPS installed at EIA. The first of these surveys was chosen because of its relative simplicity in methodology and procedures. It is expected though, that much of the learning derived through the conversion process will be useful in loading the second, and following surveys. The edit and imputation requirements of the first survey are straightforward and similar to other surveys, even though reduced and simplified. This paper will focus on those two processes for this survey.

Editing and Imputation Requirements

The edits required by the first survey, the EIA-64A, being implemented in StEPS are classified by severity as either fatal or warning (query edits). The edits designated as fatal are of three types--invalid, null, and positive whole number. Within the fatal category, ten reported items are required to be checked. Of the these items, two items are checked for both invalid and null, and two other items for both null and positive, whole numbers. Three of the reported survey items are what StEPS refers to as roster items.[1] In addition, to the 10 reported survey items checked, there is also a fatal error check on the ratio of two items being valid, so that the total number of reported items involved in the fatal edits is twelve. During implementation the requirements were revised to include edits on two more items, reported totals, for all three edit types, invalid, null, and positive whole number.

Fatal edit failures are unacceptable for processing. Warning edit failures, on the other hand, are used to indicate questionable data for review and follow-up, but the data can be used for processing. Six items are involved in the warning edit checks, of which two are roster items. These edits include three types: 1) item or item ratio compared to a constant, 2) item change with respect to previous period, and 3) item difference with respect to other survey data. During implementation these requirements were also expanded to include checks on two items equaling the sums of reported items. Some edit warnings were described in the original requirements but are enhancements to the current operating system. These edits involve comparison of items to the period two previous (t-2), and item comparisons to other survey data. These warning edits, however, were dropped from the revised implementation requirements and designated as a later enhancement. Even though these edits failures are warnings, regardless of if the failed data are to be used, the requirement called for the user to key in a code that describes why the flag is over-ridden and the data accepted or not. These reason codes include verification with the respondent, notes provided on error resolution, analyst's judgement, trend analysis support of reported data, no valid alternate source data for imputation or estimation. The reason codes associated with edit failures that result in overriding the reported data include

[1] Roster items are items which have various categories that are not hard coded on the survey form. The respondent enters a code that defines the item to be reported and then the value for that item. The code is predefined but not preprinted on the survey form. For example, expenditures by North American Industrial Classification code or, in the case of this survey, liquids produced by area of origin, are roster items.

respondent error, amended survey form received, or other survey comparison update.

Because surveys evolve as changes are made in the industry , there was a general requirement that the system allow select users to add, modify or delete rules and parameters. It was also required that the system be able to produce reports, printable and viewable, that list the edit failures and that the system provide the ability to filter and sort within the reports by different criteria. Reports should contain the flagged data, resolved and unresolved, error codes and descriptions, and correction codes. In addition to detailed level reports, summary reports were also required.

The imputation requirements were defined more broadly in terms of the system providing an automated imputation routine that produces imputed values that are stored in the database along with the original value, amended value, and estimated value. The user would also be allowed to enter manually impute a single cell value or an entire form. The requirements also specified that the imputation process would be independently initiated, and not automatically initiated. The only particular imputation method specified was imputation using the sum of "like items" from a monthly survey to produce annual totals, and apply conversion factors for appropriate units of measure. In addition, manual imputation has to be allowed such that a user can create estimated values that are differentiated from other values. Reason codes describing why data are estimated are required to be stored with the data. Reasons for modifying the data include respondent error (wrong units of measure, wrong line reported, misunderstood directions, etc), amended form received, and update based on comparison to another survey. Similarly, reason codes are also required when data are imputed. These include reasons such as no data received, and data imputed based on analyst's expertise.

The requirements for performance measures for this survey are very basic and include respondent level tracking reports and summary counts of total edit failures, resolved failures, unresolved failures, and resolved failures by correction code. Imputation performance measures requirements are only tracking in nature (reason codes, method, etc.); summary counts by type, group, etc., were not specifically included in the requirements for this survey. Such summary measures will however be required of other surveys intended to be implemented in StEPS.

StEPS Approach

StEPS currently provides for five general types[2] of edits: 1) required data item (tests if an item is missing), 2) range test edit (tests if an item falls between an upper and lower bound), 3) list directed (tests if an item equals a value in a discrete set), 4) balance (tests if the total equals the sum of detail items and flags the total if the test fails), 5) survey rule (tests specified by the user using SAS code such as item to item comparison). The user selects for the each edit test the status of the test--active, inactive or pending. Three of the edit types-- list derived, balance and survey rule-- require that the survey rule tests validate that the expressions are valid, rather than the actual data item values, but still identifies a specific "review item" when failed. The error condition is specified in SAS code. For these three types, syntax checks and copy code buttons are provided. To further assist the user, default SAS code is provided, except in the case of the survey rule edit. No edit definition can contain both roster and non-roster items. All of the five edit types, except the list derived edit, allow a pre-condition to be set to define/restrict the situations for executing the edit. The user is provided the ability to have the syntax of the pre-condition checked immediately, and to copy code from a test already in place. Each edit type is applied to whichever events the user chooses: a single ID, a pre-edit, or full-edit.

Single ID event edits are usually interactive, while pre-edits and full-edits are performed in batch mode. Pre-edits are simple edits usually run early on in the survey process that do not depend on how much data has been received, while full-edits are more commonly run after most of the data has been received. Most often, a more restricted set of users are provided privileges to run the full edit on the entire file.

Batch edits, also referred to as survey level edits, are run using scripts. They can be run immediately (in which case no other processing can occur in StEPS), immediately but in the background (StEPS processing can continue), or at a user scheduled day and time. The user also specifies which of the five types of tests, or all, to run and whether to run on all IDs, or a particular selection set. The edit failures are placed on a survey level reject file. The appropriate respondent ID record on the stat period control file is updated to reflect at least one edit test failed (F) or all tests passed (P) the full-edit.

On the other hand, interactive edits, also referred to as user level edits, are run immediately on a selected ID for one of the five types of test, or all tests. In the case of interactive edits, the failures or rejects are written to the user's own file, separate from the survey level failures,and do not affect other users. If a user has appropriate privileges, the user can add or modify the interactive edit definitions through survey specific screens.

[2] Two additional types, skip pattern validation and negative edit, are planned but currently not available.

Table 1. Batch and Interactive Edit Characteristics

Characteristic	Batch Edits/Survey Level	Interactive Edits/User level
Run Privileges	"P" (more restrictive)	"U" (less restrictive)
Run from	Run Process module	Run Process module
Run on--which respondents	All or selection set	All or selection set
Run on--which edit types	Select one or all	Select one or all
Run on--which events (pre, full, single)	Pre-edit, Full-Edit (including other script edits)	Pre-edit, Full-edit
Run when	Immediate, Immediate Background Schedule day time later	Immediate
Notification of Edit run completion	E-mail if selected	Pop-up notification
Results stored	Editrej in data file	Editrej in user library[3]
Id Flag in Review and Correction	S (survey level)	U (user level)
Identification in Review and Correction	EDTPF= F (fail) , P (pass)	selection set SELSET_E
Summary Flag in Stat Period Control File	F (at least one edit test failure), P (no failures)	

In all three types of events (pre-edit, full-edit, or single ID), the edits only identify the failures, they do not change the data. However, for four of the five edit types (the balance edit is excluded), the user can also choose to select the "g-event", in which case if the edit test is failed, the item will be marked for imputation in the general imputation module. Table 1 summarizes some of the characteristics of batch and interactive edits.

StEPS provides the user the ability to review or list/print edit failures, or review/print summaries of the results. The user selects either survey level or user level results. Review of failures for the level chosen presents all failures by ID, by error sequence, error type, the descriptions on the failure condition, run date and user. Within that screen, a summary count of IDs in error and the total number of errors is provided. The option to list failures provides mostly the same information in a more compressed layout excluding the summary counts. The review summaries option provides for the selected file, user or survey level, the count of failures for each individual test. The type of edit is also displayed for each test. A different module, Review and Correction, is used to correct item values. The ID by Item Data Review and Correction screen contains both control information and item data. One of the control data fields, Bypass, is relevant to edit and impute because it controls whether an ID should be included in processing. Here the user indicates if the ID is to be edited, simple imputation performed on the ID, general imputation performed on the ID, include the ID in the imputation base, and whether to replace data through batch update. The ID Flag field displays if there are survey level edit rejects(S), user-level edit rejects (U), or Single ID-edit rejects (W). Clicking on the S, U, or W accesses the rejects. For the item data information, each survey item is a row identified by the item number. The current period's reported value and one previous value and the current to prior ratio, or two previous values are then listed. Each value is immediately followed by the Data Flag which identifies the source of the edited data item value. The flags include sources: analysts correction, respondent change to prior period data, derived data treated as reported, edit failure impute treated as reported, edit failure treated as reported, and edit failure impute treated as imputed, historic change, analyst impute, delinquent impute, respondent reported data, other source treated as reported, other source treated as imputed. Some of the flags are set by the analyst, and some are set by the imputation program or the batch update program. Some are treated as reported and some

[3] Over-written each time interactive batch is run.

as imputed. Corrections to data items are made by typing over an existing value or deleting the value and entering the new value, given that proper privileges are set (U or P). The correction then require that the data flag be set or accept the default of analyst correction. After all changes have been made and the user applies the corrections, the audit trail is updated. The audit trail includes for the ID, and item, the old data flag and new data flag, and the old data value and new data value, the user and time of the update.

StEPS has two types of imputation–simple and general. Within simple imputation, there are two types available–free form and balance complex. Simple imputation usually takes place prior to editing, and is often used to fill in data not provided that can be inferred from the provided data. Free form imputation requires the user to specify the error condition and action to be taken in SAS code in an if-then format to define the when and how for imputing. Balance complex is pre-coded but requires the user to specify the complex, the detailed items and their total, relative tolerances on the absolute residual and/or the relative absolute residual, and select one of ten pre-defined conditions and actions. The conditions include, for example, more than one detail item is missing but not all, the total is greater than 0, the sum of the detail items is greater than 0, but the total is less than or greater than the sum of the detail. The corresponding action is to set the total equal to the sum of the detail. Data values imputed in simple imputation are considered to be equivalent to reported data. The imputed data are flagged as being reported, and the method used recorded in the imputation flag field.

In comparison, general imputation is usually executed after editing. General imputation flags the changed values as imputed data. In general imputation, the user determines for the survey whether to impute for non-respondents or perform non-response adjustment instead. Users provide specifications as to the order in which to impute items, which methods to use under certain conditions, etc. Table 2 shows the methods for imputing individual items. The user also specifies the actions for adjusting balance complexes. As in simple imputation, tolerances are set by the user for each method within balance complex. In the current version of StEPS, roster data are not subject to general imputation.

The Impact Flags for each non-roster data item are displayed through the Review and Correction module described above for the editing review screens. The values indicate whether the imputation rules for this ID were followed, imputation performed regardless of the rules, imputation was not performed even though rules indicate impute, or rules were followed for imputation but the ID was excluded from the imputation base. In addition, the Imputation Flag is displayed. These flags are set by the imputation program to identify the method of imputation used, the type of imputation, if in the imputation reject file, and if used in the imputation base. Roster data, however, can be viewed only through the roster item matrix screen. This screen is also used to edit roster item data for a single ID or perform roster item data simple imputation for a single ID.

The MIS module of StEPS contains Edit Summary and Imputation Rates for performance measures. Edit rates, viewable through the Response Rates file, are calculated based on the number of cases edited at the survey-level (run via the script option) and the number of all active cases. Also, through the MIS module, the production log can be read to determine start times, end times, elapsed time, number of observations processed, in particular for executing batch jobs for general imputation or editing.

Comparison of Requirements and StEPS Features and Functions

StEPS appears to accommodate most of the specific edit rules of the survey first tested, the EIA-64A. However, there does appear to be some gaps between the current survey requirements and the StEPS editing process. These gaps relate to the editing of roster data, levels or priorities of edit failures, and business rules and process control. As is common with many EIA surveys, much of the test survey data reported is open-ended. Because a respondent is likely to only need to report for a few of a large number of potential categories, respondents fill in the category, and the value for that category. This approach reduces the size of the form and eliminates the many cells that would otherwise be empty on the report. This type of data where the respondent reports the category and the value is referred to in StEPS as roster data. Roster data are a later addition to StEPS, and, as a result, roster data are treated differently than non-roster data. The definition of editing rules for roster data is separated from non-roster data. Furthermore, roster data can't be combined with non-roster data in editing rules. Similarly in imputation, roster data has separate imputation definitions for simple imputation, and appears to be limited to only balance complex. At this time, there is not a roster general imputation module. The other limitations placed on roster data and estimates based on roster data and performance measures are still being discovered by EIA, and resolutions sought.

The second main area of difference found between StEPS and the requirements of the test survey is edit failure levels or priorities. A number of EIA surveys use the concept of critical, in addition to the fatal and warning levels required by the test survey, to distinguish levels of severity, and to prioritize the order of the work involved in resolving edit failures. Across EIA surveys, warnings are almost universally defined as query edits.

Table 2. Item Imputation Methods

Method	Description	Formula
ATREND	Use auxiliary variable adjusted by trend	$x' = z_1 (z_2/z_3)$
AUXRAT	Use auxiliary variable adjusted by ratio of identicals	$x' = z_1 (S(z_2) / S(z_3))$
MEAN	Use the mean of auxiliary variable	$x' = \sum z_i/n$
MULTREG	Use multiple regression prediction from auxiliary variables	$x' = \beta_1 z_1 + \beta_2 z_{2+} \ldots \beta_n z_n$
PRODUCT	Use the product of two auxiliary variables	$x' = z_1 z_2$
RATIO	Use ratio prediction	$x' = (s(x) / S(z_1))_I z_1$
RESIDUA	Use auxiliary variable minus the sum of other auxiliary variables	$x' = z_1 - _{(} z_{2+} \ldots _{+} z_{n)}$
SIMPREG	Use simple regression prediction from auxiliary variable	$x' = \beta_1 z_1$
SUM	Use the sum of auxiliary variables	$x' = z_1 + z_{2+} \ldots _{+} z_n$
VALUE	Use the value of an auxiliary variable	$x' = z_1$

Edit failures in this category are flagged to be reviewed, but the data are not considered to be necessarily wrong. The outcome of the review is to accept, change, or mark the value to be replaced by an imputed value for the purpose of aggregation/estimation (the reported value currently is always preserved on the file though). Fatal and, in some systems, critical errors, on the other hand, do not have accept as an allowed outcome. If the error is not corrected or marked to be overridden with an imputed value, the value can not be used for processing. The StEPS functionality appears to easily satisfy requirements for warning edits, including reason codes to document the process but does not assign priorities among the edits. In addition, it also appears that fatal error requirements of the test survey could only be partially accommodated though the impute indicator set in the edit definitions and passed to the General Imputation module. Also, the functionality provided through the item dictionary "required" flag could be used to determine the response not a valid response, and therefore classify it as a nonresponse, but this would apply to the entire form. These two functions could prevent the fatal type failures or the entire form from being used by automatically replacing the value(s) with an imputed value, but does not provide the mechanism for prioritizing fatal failures for manual resolution that require correction before aggregation/estimation can be performed.

The third area of difference relates more generally to business rules and process flow and process control. One of the business rules followed in processing EIA survey data is that each response is processed using the same edit rules, regardless of the response mode, the data analyst, etc, with a complete audit trail. Batch edits in StEPS are consistent with this principle. Batch edits in StEPS are run according to a selected time schedule on the entire survey, and the rejects are stored in the Data Library. These rejects are survey-level. They can be viewed through the Edit Results Screen or the Review and Correction module through the creation of a selection set. However, StEPS treats interactive edits differently than batch edits. Interactive edits can be run for the entire survey or just on a selected set of respondents. They are run immediately from a different module than for batch edits, but do not impact other users working with the survey data. The edit rejects are stored in a different library than batch edit rejects, the Userlib library. These rejects are user-level and are overwritten with new rejects each time an interactive edit is run. The different treatment of batch and interactive edits as survey-level and user-level diverges from the current survey principles and practices. It is not evident that the StEPS interactive edits as designed when used for production, rather than testing, would produce a systematic, reproducible result from period to period, and data analyst to data analyst. EIA's current production systems have both batch and interactive edits processes that are mirror images.

Interactive edits are used to quickly process late respondents or to re-edit data that have been revised since the initial edit. The edit rejects are permanently stored in a manner identical to batch edits rejects. It is not apparent how to duplicate that requirement in StEPS, given the separation of files and the temporary nature of user-level rejects. Possibly the solution is to only do batch edits, the majority of which are scheduled for later, but for late respondents and re-edits schedule now. On the other hand, the interactive edits at the user-level perform a function and flexibility that doesn't exist in the test survey currently or most of EIA's current production systems. That type of function currently would more likely be performed offline for unique cases, but wouldn't be considered part of editing. It does provide the opportunity of testing new edits or performing post-batch edit analysis. More broadly though, the general area of concern in using StEPS is the overall process flow and control. EIA's current processing systems' editing modules are only executed by the assigned programmer. Output is then produced (hard copy or electronic) for data analysts to examine, and resolve edit failures. While the edit resolution process in StEPS appears similar, the process flow leading to the resolution, edit failure identification, is a more open process, because of the ease of changing edit definitions and running script files, and the capability for user-level files. While this process can be somewhat controlled through the assignment of user privileges, it is not yet evident that will be sufficient. Again, this same issue can also be viewed as an enhancement that makes the survey process less linear, with capabilities for analyst to create/modify edit definitions in a timely manner and discover data errors that might otherwise be undiscovered.

StEPS provides imputation methodology capabilities with scope beyond most EIA processing systems. In particular, current systems provide one method of imputation. The one method however varies across surveys. The test survey has simple requirements, to use the same respondent's data from another survey. When that doesn't exist, imputation is performed using the data from the same survey's prior period. It is also required that manual imputation based on an experts opinion be allowed by the system. StEPS though provides the user both simple and general imputation. In simple imputation, the user is provided multiple choices of imputation, particularly through free form, but also sophistication within balance complex.

Balance complex imputation is used when either the total of a group of items, or one or more detailed items within a group is missing, given that the available data are complete enough for imputed values to be considered equivalent to reported values. If only the total is missing, balance complex imputation can set it to the sum of the detail. If only one detailed item is missing, balance complex imputation can set it equal to the total minus the sum of the reported details. If the total is not equal to the sum of the detail, the user can chose (RAKE) to adjust the details, and preserve the reported total. The reported detailed items (x_i) are adjusted proportionally, so that the adjusted detailed items are: $x_i' = x_i(y/\sum x_i)$, where y is the total of the details. Despite the variety of methodology for imputation though, StEPS does not allow general imputation for roster items. A significant amount of the data EIA collects, both in the test survey and other surveys, are roster items. General imputation is a necessity to impute for edit failures and nonrespondents. To resolve this issue, it may be required to change the general imputation module to convert roster data and create a roster fat record similar to non-roster data.

Conclusion

StEPS provides the ability to satisfy most of the editing and imputation rules required for the test survey, but there does appear to be some gaps between the current survey requirements and the implemented version of the StEPS editing and imputation processes. These gaps relate to the editing of roster data, levels or priorities of edit failures, and business rules and process control. As testing and implementation of StEPS at EIA continues, it will be determined if these gaps can be closed with revisions to the current version, changes in EIA's approach, or if they will prevent EIA from fully implementing StEPS for production for multiple surveys.

References

[1] Sigman, Richard (2001). "Editing and Imputation in a Standard Economic Processing System", Richard Sigman, *Proceedings of Statistics Canada Symposium 2001, Achieving Data Quality in a Statistical Agency: A Methodological Perspective.*

[2] "EIA-64A Survey Processing System, Requirements Specifications Document", internal EIA document (2001).

[3] StEPS User Manual, http://www.census.gov/esmpd/www/steps/documentation.htm