APPLICATION OF THE SPEER EDIT SYSTEM

William E. Winkler and Lisa R. Draper
U.S. Bureau of the Census

ABSTRACT

This document describes an application of the SPEER (Structured Programs for Economic Editing and Referrals) edit system.

Keywords: Fellegi-Holt, localization, imputation

This document provides background on the workings and an application of the SPEER (Structured Programs for Economic Editing and Referrals) edit system. The first three sections consist of a description of the basic edit system, an overview of how to develop and run a SPEER edit system, and an example showing specific details of the input and output files used by the software. In the fourth section, we present an application to the Census of Manufactures, the largest U.S. economic survey. The final section is a summary.

I. DESCRIPTION OF THE SPEER EDIT SYSTEM

The following subsections describe aspects of the SPEER edit system.

1. Purpose, Model, and History

The SPEER edit system is designed for ratio edits of continuous economic data. The system utilizes the Fellegi-Holt model of editing. The first version of SPEER was written by Brian Greenberg (Greenberg and Surdi 1984, Greenberg and Petkunas 1990) and the current version was written by William Winkler (1995). The computational algorithms, much of the imputation methodology, and the source code in the current version is new.

2. Software and Computer Systems

The software consists of two programs, gb3.for and spr3.for. The software is written in portable FORTRAN which should recompile on a variety of computers. It currently runs on IBM PCs under DOS, Windows, or OS/2, DEC VAXes under VMS, DEC Alpha under Windows NT, UNISYS, and a variety of UNIX workstations. The programs run in batch mode and the interface is character-based.

The first program, gb3.for, generates the entire set of edit bounds. The main input is a file containing at least three lines. The first line is the name of the input file of explicit edits, the second is the name of the output file of implicit edits, and the third is the name of the output summary file. If a fourth line is present, it consists of the FORTRAN FORMAT for the input file of explicit edits. If a file of variable names, BNAMES.DAT, is present, then the variable names in it are used; otherwise, default names of the form VRnnn are used where nnn can range as high as 999. After the

main input is read, the inputs and outputs are the usual ones associated with edit-generation programs. The most important input is the file of explicit edits that have been defined by an analyst. This input must be in a fixed format that is specified in the program documentation. As output, gb3.for produces the file of implicit edits that are logically derived from the explicit edits and also checks the logical consistency of the entire set of edits. With appropriate test data, an auxilliary program D-MASO (also in FORTRAN) can help an analyst determine the lower and upper bounds on the ratios that are in the set of explicit edits. The appropriate test data might consist of prior year's edited data or (a subset of) the current year's data.

The second program, spr3.for, performs error localization (i.e., determines the minimal number of fields to impute for a record failing edits) and then does imputation. The main input is a control file with at least five lines. The five lines are (1) the name of the input file being edited, (2) the name of the file containing implicit edits, (3) the name of the output, (4) the FORTRAN format of the quantitative data in the file being edited, and (5) the number of variables (fields) being edited. Five additional lines are also read in. They are (6) the name of the file containing implicit edits, (7) the name of the file containing variable names, (8) the name of the file of beta coefficients, (9) the file of weights, and (10) optional FORTRAN FORMAT for file of explicit edits. The first six lines are mandatory. If the last five lines or the associated files do not exist, then defaults are used. The weights affect which fields are imputed. The variables with lower weights are imputed before those with higher weights. After the control file is read, the input files consist of the set of implicit edits produced by gb3.for, the data file being edited, and a set of "beta" values associated with ratios. The beta values are determined a priori using an appropriate test deck and consist of regression coefficients under the model $y = \beta x$. There can be as many coefficients as there are implicit edits. The imputation methodology consists of first determining an imputation range for a variable so that edits are satisfied. Within the range, the first choice of imputation uses a reported variable that is not being imputed and the corresponding "beta" coefficient. After the first choice, a hierarchy of defaults based on the imputation range is selected. Regression imputation is only used when the appropriate beta coefficient is available and the variable being imputed is associated with a variable that is reported. By the Fellegi-Holt theory, any values of fields chosen in the imputation range necessarily yield complete multivariate records that satisfy all edits.

The outputs from the second program consist of summary statistics, the file of edited (i.e., containing imputes) data, and a file giving details of each record that was changed. The details consists of the failed edits, the minimum fields to impute, and the imputation methodology that was utilized for each field.

3. **Documentation**

Three documents describe the overall SPEER methodology and capabilities. They are Greenberg and Surdi (1994), Greenberg and Petkunas (1990), and Greenberg, Draper, and Petkunas (1990). The documents do not describe details of the algorithms or how to create and run the system for specific data bases. New computational algorithms (Winkler 1995) eliminate much of the redundant computation of earlier versions of the SPEER system. Major restructuring of the computer code makes the system much easier to apply in new situations because only one FORTRAN FORMAT statement describing locations of input fields in the file being edited must be changed. Winkler (1994) describes how to develop and run a SPEER system.

Documentation related to the details of the software and how to run the software has been created

for the first time (Winkler 1995). The main documentation consists of instructions on how to run the example that is included on the disk with the software. Each program has internal documentation (in comments at the end) describing the nature and structure of the inputs and the outputs. The internal documentation should be sufficient to allow all but the most naive users to apply the software in a variety of situations. The new source code is more easily understood because of its modular structure. In most applications it is unlikely that source code (with the possible exception of two parameters that determine that amount of allocated storage) will need modification.

4. **Limitations**

SPEER only deals with ratio edits. For a new user, the file of explicit edits may not be very easy to develop. A statistical package should be used to determine those variables that are linearly related and the associated regression ("beta") coefficients. The regression model is $y = ß x$. Those "beta" coefficients that are placed in an external file are used for the default imputations. If "beta" coefficients are not available for two variables that are associated via a ratio edit, then the default imputation is based on allowable range that satisfies the edits. The best imputations require survey-specific modifications in which the imputation module is replaced by special code.

The main output from spr3.for is a large print file that contains details of the failed edits, the error localization, and the imputations that were made. The program spr3.for does not produce an output file that has the same FORMAT as the main input file being edited and that has appropriate quantitative data (missing or edit-deletes) replaced by imputations. This is not done due to the difficulty in writing necessary generalized i/o routines, documenting the routines, and getting users to understand how to carry and output additional information from the input file that does not pass through SPEER edits. The program spr3.for does produce an output file EDIT.OUT that contains all the quantitative data fields that pass through the edits and that contains the newly imputed values. It is output in a fixed format and could be merged in with the original data that passes through the edits because it corresponds on a line-by-line basis.

The program spr3.for does not impute values for variables in connected sets in which all values are blank. A set of variables is connected if they are connected via ratio edits. Connected sets form a natural partition of the entire set of variables being edited. If all variables in a connected set are missing, then imputation cannot be based on ratios and must be determined via default procedures that might possibly be based on data from a prior time period.

5. **Strengths**

The software is very easy to apply because only one format statement describing the locations and sizes of the quantitative being edited needs to be changed (Winkler 1995). In situations where storage does not exceed the default storage of the program, the FORTRAN format statement can be read in from an external file. Thus, the software does not need to be recompiled when it is used on different data files. While the software will handle a moderately large number of variables (200+), the present computational algorithms, with suitable modification, could allow it to handle more than 2000 variables. The software is fast. For instance, to generate 272 pairs of implicit edit bounds in each of 546 industrial categories for the Census of Manufactures requires only 35 seconds on a Sparcstation 20. Because ratio edits are basically simple, algorithms and associated source code are quite straightforward to follow or modify. For most situations, source code should not need any

maintenance or modification. All core edit algorithms are in debugged code that is reusable. Checking the logical consistency of the set of edits (via gb3.for) does not require test data. Default imputations are quite straightforward to set up. A new software program cmpbeta3.for will compute the "beta" coefficients for all pairs of variables (fields) that are associated via the ratio edits that are explicitly defined. The program cmpbeta3 is approximately 50 times as fast as commercial software because it contains no diagnostics or special features.

II. DEVELOPING AND RUNNING A SPEER EDIT SYSTEM

   This section provides an overview of how to create and run the SPEER edit system. It describes some of the non-SPEER components that must be used in addition to the SPEER components. It also gives the type of personnel that are useful as an edit team developing a system.

1. Developing an Edit System using SPEER

   There are three facets to the development, (1) analysis of the data using statistical and other packages, (2) development of a pre-edit system, and (3) development of a SPEER system. If data from a prior time period are not available, then data obtained during the collection can be used.
   Stage 1 proceeds with a variety of steps. The analyst would begin by running various tabulations on the data to determine means, variances, ranges, and other values. Next the analyst would run a regression package to determine which continuous variables are linearly related and to get a variety of diagnostics. The pairs of variables that are linearly related and the associated "beta" coefficients from the regression need to be stored. When data from a prior time period is available, then analysts often have much of this information already.
   Stage 2 consists of preliminary edits that often do not require sophisticated rules. These can involve checking whether a State code takes a value within a set of correct values, a variable takes a value in a specified range, and a group of variables adds to a desired sum.
   Stage 3 begins with determining the edit bounds for ratios. To facilitate the process, we have a software tool, D-MASO, developed by David Paletz, that delineates potential bounds and a variety of diagnostics. The analysts can then quickly determine bounds. SPEER software consists of two components. The first, gb3.for, generates the logically implied edit bounds and checks the consistency of the entire edit system. It does not require test data. The second component consists of the SPEER edit, spr3.for. It determines edit failures, the minimum number of fields (variables) that must be changed so that the record satisfies edits, and then does imputation. The first program only needs the set of explicit edit bounds as input. The second program needs the set of implicit edit bounds from the first program, the set of "beta" coefficients from the regressions, and the data file that is being edited. A new program cmpbeta3.for will compute the "beta" coefficients for all pairs of variables that are connected via ratio edits. The program requires the file of explicit ratio edits, the main file being edited, the FORTRAN format of the quantitative data in the file being edited, and the number of variables (fields) being edited. It computes beta coefficients for all pairs of variables that can be associated via implicit ratio edits.

2. Maintenance of SPEER Code

   The code may not require any maintenance. If larger data structures are needed, then the two

parameters at the beginning of the code should be changed and the program recompiled. If the imputation module is changed or a new one is developed, then updating merely involves substituting the new subroutine for the old.

The code is very modular and contains much internal documentation. In particular, comments at the end of the code give details related to running the programs.

3. Other Maintenance of a SPEER System

The analyst must document how the "beta" coefficients from the regressions are obtained. The program cmpbeta3.for can quickly produce the set of "beta" coefficients.

4. Edit Team

An edit team is most useful when it consists of at least one individual in each of the following categories: (1) methodologist, (2) analyst, and (3) programmer. Development of an edit system is primarily a programming project once subject-matter and analytic needs are identified.   The methodologist could be an economist, demographer, or statistician who is familiar with the Fellegi-Holt theory and can facilitate the programming of the system. The methodologist can provide an important focal point if the methodologist can make sure that programmers are given knowledgable information about system requirements and understands details of programming such as how long it takes programmers to develop new, difficult skills. The analyst is a subject-matter specialist who is familiar with the industries for which data are being edited. Often analysts and programmers have worked together successfully on other projects. Teams often start slowly because of the time needed to develop common terminology and communication skills. Once team members are working closely together, however, final products are often better because individuals are stimulated by detailed knowledge provided by other team members.

III. EXAMPLE

The example basically shows what the inputs and outputs from running the two programs of the SPEER system look like. The first program generates all the implied edits that are needed for error localization and checks the logical consistency of the entire edit system.   An edit system is inconsistent when no data records can satisfy all edits. The second program uses the entire set of edits that are produced by the first program and edits data records. For each edit-failing record, it determines the minimum number of fields (variable values) to change to make the record consistent.

1. Implicit Edit Generation

The first program, gb3.for, takes a set of explicit edits and generates a set of logically derived edits. The edits consist of the lower and upper bounds on the ratios of the pairs of variables. Two tasks must be performed. The first is to create an input file of explicit ratio bounds. The bounds are generally created by subject-matter analysts who are familiar with the survey. An example is given in Table 1.   The eight fields of the input file are: form number, edit-within-form-number, variable number of numerator, variable number of denominator, lower bound on ratio, upper bound on ratio, an intermediate value between the lower and upper bounds, and the four-character names of the

variables. The form number describes the industry to which the edit refers. With U.S. Bureau of the Census surveys, the same form may be sent to all companies over a broad range of industrial classification categories. Separate ratio bounds need to be developed for each industrial classification.

Table 1. Example of Explicit Ratio Bound Input File

```
110   1   1   2        .0212400        .0711125        .0369900   EMP1/APR2
110   2   2   3       1.5369120       6.8853623       3.2590401   APR2/QPR3
110   3   3   2        .1670480        .5273000        .3068400   QPR3/APR2
110   4   4   2        .0202880        .2717625        .0929800   FBR4/APR3
```

The second field refers to the edit number. It is primarily for the benefit of the analysts and is not used by gb3.for. The next two fields are the variable numbers of the fields in the ratio and the following two are the lower and upper bounds created by the analysts. The final two fields are not used by gb3.for but can be used by the analyst. The next-to-last field is possibly an average or median value that the analyst enters in the input file. The last field is a character representation that helps the analyst remember the variables. For instance, QPR3 might refer to "quarterly payroll" and APR2 might refer to "annual payroll."

The second task is only needed if default storage allocations are not sufficient. The task requires changing a parameter statement at the beginning of the program and recompiling the program. The statement has the form

PARAMETER (BFLD=45).

BFLD refers to the upper bound on the number of variables (here 45) being ratio edited. The number of variables being edited is assumed to be the same in every industry if more than one industry is edited. For the example, the output file primarily contains the ratio bounds (implicit edits) for the six pairs of the four variables.

2. **Error Localization**

The main edit program, spr3.for, takes three inputs. The first is the set of implicit edit ratios produced by gb3.for. The second is a set of "beta" coefficients that are created by a regression package that the analyst has used. The third input is the file being edited. A FORTRAN FORMAT statement that describes the locations of the input variables in the third file must be modified and placed in an external file. A parameter statement at the beginning of the program

```
PARAMETER(BFLD=45,BCAT=3,NCENVL=BFLD,NFLAGS=9,N_FLG=100,
+ NEDIT=BFLD*(BFLD-1)/2,MATSIZ=BFLD)
```

must also be changed. BFLD and BCAT are upper bounds on the amount of storage that is allocated. NFLAGS and N_FLG are upper bounds on storage for errors for a single record. In many situations, the default values of these parameters will be sufficient. If they are not, then parameter values will need to be increased and the program must be recompiled. Comments at the end of the source code give many details of setting up and running the program.

Two output files are produced. The first consists of summary statistics. The second (see Table 2) contains details of the edits, blank fields, and imputations for each edit-failing record. The output shows what edit has failed, the minimum number of fields that must be imputed, the imputation method that was adopted, and the revised and reported values of the record.

The program spr3.for is set up so that a more sophisticated imputation can easily be substituted for the existing one. Basically, analysts would have to do more modelling and determine a hierarchy of imputations that would be coded in a subroutine. The imputation subroutine would be added to the code and the eight lines associated with the existing (default) imputation would be replaced by a call to the subroutine. Documentation in the code clearly shows where the substitution should be made and what data must be passed to and from the imputation subroutine.

Table 2. Example of Edit-Failing Record in Main Output from SPR3.FOR

```
  Record #   1

Failed edits:
       1.8964540 <  APR2  /  QPR3  <        5.9863030

Deleted fields:    3. QPR3

Imputation range for  QPR3 :  Lo =            3.3410    Up =            10.5460
   QPR3  imputed using  QPR3 / EMP1  ratio

    Fields      Revised      Reported       Lower        Upper
    ------      -------      --------       -----        -----
     EMP1         1.000         1.000        .425        1.422
     APR2        20.000        20.000      14.062       34.207
     QPR3         5.714        13.000       3.341       10.546
     FBR4         3.000         3.000        .406        5.435


  Record #   5

Failed edits:
        .0402807 <  EMP1  /  QPR3  <         .4257010
       1.8964540 <  APR2  /  QPR3  <        5.9863030

Deleted fields:    3. QPR3

Imputation range for  QPR3 :  Lo =            6.6819    Up =            21.0920
   QPR3  imputed using  QPR3 / EMP1  ratio

    Fields      Revised      Reported       Lower        Upper
    ------      -------      --------       -----        -----
     EMP1         2.000         2.000        .850        2.845
     APR2        40.000        40.000      28.124       68.415
     QPR3        11.429         4.000       6.682       21.092
     FBR4         6.000         6.000        .812       10.870
```

IV. APPLICATION

SPEER is currently being used in two large interactive applications. These applications are the Annual Survey of Manufactures and the Census of Manufactures and Mineral Industries. The applied system, named LRPIES (Late Receipts Processing and Interactive Edit System), is used primarily for basic data entry and editing, editing of late receipts, and processing establishment adds. The current version has features that facilitate analysts' review and correction of data records. Analysts in Washington can now enter and correct late receipts that arrive after the central data processing center in Jeffersonville, Indiana has shut down. Previously, late data were entered but generally left unedited. Analysts can also perform additonal review of the non-late data that were previously edited at the Jeffersonville location.

The SPEER application (LRPIES) involves the largest U.S. surveys of industry and manufacturing. As much analyst review of data is needed, custom software modifications that provide assistance and

review capability have been added. The modifications are specific to Digital VAXes and the large screen display capabilities of the types of VAX terminals in use. Records that have failed edits and that require imputation to make them consistent with the set of edits can be retrieved and processed interactively. For each edit-failing record, a number of values are displayed that facilitate the analysts' review and correction. The values are current values, a prior time period's corresponding values if available, suggested impute values, and ranges in which values can be imputed that are consistent with the set of edits. Analysts --possibly after a call-back-- have the capability of entering a flag that causes an edit-failing value to be accepted. The custom code in LRPIES associated with the interactive edits is the majority of the code. The main SPEER subroutines merely need to be called and do not need to be modified.

   The LRPIES application needs edit parameters and information for 546 SIC (Standard Industrial Classification) codes. The main edit parameters are the lower and upper bounds associated with the ratios being edited. Bounds from a prior year are often used as the starting point in producing the bounds for the current year's edits. Edit bounds and information can vary substantially across SIC codes. The specific parameters and information are the implicit edits for the current year and the prior year, the industry average value, and the beta coefficients obtained from regressing one of the variables (fields) in a ratio against the other variable. While the basic SPEER imputation merely uses a regression imputation, the LRPIES application uses a hierarchy of imputations based on the existence of prior data. The exact types of imputations and the hierarchy are determined by analysts familiar with the data.

V. SUMMARY

   The SPEER system is a Fellegi-Holt edit system for ratios of linearly related data. It is written in portable FORTRAN, easily applied, and very fast. Applications of SPEER include some of the largest U.S. economic surveys.

REFERENCES

Greenberg, B. G., Draper, Lisa, and Petkunas, Thomas, "On-Line Capabilities of SPEER,"
 presented at the Statistics Canada Symposium, October 1990.
Greenberg, B. G., and Surdi, Rita (1984), "A Flexible and Interactive Edit and Imputation System
 for Ratio Edits," SRD report RR-84/18, U.S. Bureau of the Census, Washington, D.C., USA.
Greenberg, B. G., and Petkunas, Thomas (1990), "Overview of the SPEER System," SRD report
 RR-90/15, U.S. Bureau of the Census, Washington, D.C., USA.
Winkler, W. E. (1994), "How to Develop and Run a SPEER Edit System," unpublished
 document, Statistical Research Division, U.S. Bureau of the Census, Washington, D.C., USA.
Winkler, W. E. (1995), "SPEER Edit System," computer system and unpublished documentation,
 Statistical Research Division, U.S. Bureau of the Census, Washington, D.C., USA.