BUREAU OF THE CENSUS
STATISTICAL RESEARCH DIVISION REPORT SERIES
SRD Research Report Number: Census/SRD/RR-86/02

THE USE OF IMPLIED EDITS AND SET COVERING IN AUTOMATED DATA EDITING

by

Brian Greenberg
Statistical Research Division
Bureau of the Census

Recommended by: Paul Biemer

Report completed: January 17, 1986

Report issued: January 17, 1986

I. INTRODUCTION

The objective of this report is to provide an introduction to and a discussion of what has come to be known as the Fellegi-Holt approach to data editing. We present the basic procedures, discuss them, and provide examples to illustrate them. The most salient feature of the Fellegi-Holt editing method is that all fields are considered simultaniously when determining values to change on an edit failing record. This report does not cover all the details contained in "A Systematic Approach to Automatic Edit and Imputation" by I. P. Fellegi and D. Holt [FH], in particular, it does not contain proofs. We refer those interested to the Fellegi-Holt paper for additional technical features of the methods presented and for an excellent discussion of automated data editing.

In Chapter II we introduce the concept of implied edits and show how implied edits are derived both for categorical and continuous data. As will become clear, the implied edits are crucial to determine fields to delete on an edit-failing record. In addition, implied edits are valuable in their own right as an aid in the evaluation of editing criteria.

In Chapter III we show how the implied edits are used to find a set of field values to alter on an edit-failing record. It is at this stage of the methodology that one employs set covering procedures that are widely used in operations research. In Chapter IV we focus on the set covering problem in general and then show how it is applied when determining fields to delete an edit-failing record. In Chapter V we discuss two programs for editing data which are based on the methods outlined in this report. In an Appendix we include computer output from a pair of programs that (1) generate implied edits for categorical data when provided with a family of explicit edits and (2) deletes fields on edit-failing records so that the remaining fields are mutually consistent. The computer print-out in the Appendix was generated when these programs were run on examples discussed in the body of this report.

The focus of this report is on mathematical techniques for error localization. That is, procedures for detecting a subset of fields to delete on an edit-failing record such the remaining fields are mutually consistent. The subject of imputation is hardly mentioned in this report at all. Imputation rules are highly survey-specific and are usually designed by subject-matter specialists knowledgeable about the special considerations that must be brought to bear for the particular survey under consideration. The crucial point to observe, however, is that an overall imputation strategy must take into account edit constraints to avoid the imputation of edit-failing values. Imputation is discussed in [FH] for categorical data and briefly in the last section of this report in terms of the SPEER System for continuous data under ratio edits.

II. DEFINING EXPLICIT EDITS, IMPLIED EDITS, AND CONSISTENT FIELDS

A. Introduction

We let a response to a questionnaire having n reponse variables be represented by a vector $\underline{a} = (a_1,\ldots,a_n)$. Let A_i denote the range of values for the i^{th} response variable so $\underline{a} \in \prod_{i=1}^{n} A_i$. In addition, we sometimes denote the i^{th} response variable by F_i, $i=1,\ldots,n$.

Definition: An edit, e, is a non-empty subset of $\prod_{i=1}^{n} A_i$, and an edit set, E, is a finite collection of edits. If e is an edit and $\underline{a} \in \prod_{i=1}^{n} A_i$, we say that a fails edit e if $\underline{a} \in e$. (We emphasize that e is a subset $\prod_{i=1}^{n} A_i$.) We say a response vector $\underline{a} = (a_1,\ldots,a_n) \in \prod_{i=1}^{n} A_i$ is consistent if there does not exist an edit $e \in E$ such that $\underline{a} \in e$.

If $\underline{a} = (a_1,\ldots,a_n) \in e \in E$, the response vector $\underline{a}$ is considered invalid or inconsistent. The set of response combinations $\bigcup_{e \in E} e \subset \prod_{i=1}^{n} A_i$ consitutes the totality of prohibited response vectors. The set $\prod_{i=1}^{n} A_i - \bigcup_{e \in E} e$ constitutes the set of consistent response vectors.

Definition: An edit set E is said to be consistent if there exists at least a single $\underline{a} \in \prod_{i=1}^{n} A_i$ such that $\underline{a} \notin \bigcup_{e \in E} e$. That is, E is consistent if $\bigcup_{e \in E} e \neq \prod_{i=1}^{n} A_i$.

An explicit edit set is a finite collection of edits which will be the starting point of our edit analysis. These edits are usually furnished by subject-matter specialists knowledgable about the survey under consideration and able to explicitly provide families of prohibited response combinations. Each element of an explicit edit set is called an explicit edit.

Definition: Let E be an explicit edit set and f a subset of $\bigcup_{e \varepsilon E} e$. If f is not in the set E, we say that f is an implied edit.

Definition: If f and g are edits we say that: (1) f contains g if $g \subset f$ (recalling again that f and g are both sets), and that (2) f properly contains g if $g \subset f$ and $g \neq f$. If X is an arbitrary set of edits and f is an edit in X, we say that the edit f is a maximal edit with respect to X if f is properly contained in no other edit in X.

In the next two sections we discuss first categorical data and then continuous data, and show how edits can be represented, manipulated, and derived.

B. Categorical Data

In this section, all data will be assumed to be categorical, and each A_i (the range of responses to field F_i) will be a finite set.

Definition: For categorical data a normal edit is an edit of the form $e = \prod_{i=1}^{n} B_i$ where $B_i \subset A_i$ for i=1,...,n. If $B_i \neq A_i$ we say the field F_i enters edit e.

Remark: We will assume throughout that the explicit edit set provided by subject-matter specialists consists entirely of normal edits. Since an arbitrary explicit edit set can be converted to a set of normal edits, this assumption is not limiting.

Example 1: The following example is based on Example 1, in [FH]. In this simple example of edits for categorical data we will have three fields and two explicit edits. The fields are:

Field Name	Possible Codes	Recodes
Age	0-14	1
	15 +	2
	Single	1
	Married	2
Marital Status	Divorced	3
	Widowed	4
	Separated	5
	Head	1
Relation to Head	Spouse	2
of Household	Other	3

The two explicit edits are:

edit e_1

$$\{0\text{-}14\} \quad \text{and} \quad \left\{\begin{matrix}\text{Married}\\ \text{Divorced}\\ \text{Widowed}\\ \text{Separated}\end{matrix}\right\}.$$

edit e_2

$$\left\{\begin{matrix}\text{Single}\\ \text{Divorced}\\ \text{Widowed}\end{matrix}\right\} \quad \text{and} \quad \{\text{Spouse}\}.$$

Note that edits express prohibited response combinations. Writing these edits in normal form using the recodes and expressing them in the form $\prod_{i=1}^{3} B_i$ we have:

	Field F_1		Field F_2		Field F_3
e_1:	$\{1\}$	x	$\{2,3,4,5\}$	x	A_3
e_2:	A_1	x	$\{1,3,4\}$	x	$\{2\}$.

Note that F_1, F_2, and F_3 represents, respectively, Age, Marital Status, and Relation to Head of Household. The presence of A_3 (for example) in the representation of edit e_1 signifies that field F_3 does not enter edit e_1, that is, edit e_1 only involves fields F_1 and F_2.

Suppose we have the following three records:

$\underline{r}_1$ = (72, Widowed, Head) = (2, 4, 1)
$\underline{r}_2$ = (72, Widowed, Spouse) = (2, 4, 2)
$\underline{r}_3$ = (12, Widowed, Spouse) = (1, 4, 2) .

Note that record $\underline{r}_1$ fails no edits and hence it is consistent. Record $\underline{r}_2$ fails edit e_2 and record $\underline{r}_3$ fails both edits e_1 and e_2, hence both of these records are inconsistent and are considered invalid.

Definition: If E is an explicit edit set, consider all implied edits which are of the form $\prod_{i=1}^{n} B_i$, where $B_i \subset A_i$ for i=1,...,n. We call this set of edits the implied (normal) edit set for E. The elements of the implied (normal) edits set are called implied (normal) edits.

Remark: We next show how to derive a family of implied normal edits from a given family of normal explicit edits.

Definition: Let $E^* = E$, $M \subset E^*$, and k be an integer $i \leq k \leq n$, and for $e \varepsilon E^*$ write $e = \prod_{i=1}^{n} B_i^e$. The implied edit, f, is said to be derived from edit set M with generating field k if

$$f = \prod_{i=1}^{n} B_i^f,$$

where

$$B_i^f = \bigcap_{m \varepsilon M} B_i^m \qquad \text{for } i \neq k ,$$

and

$$B_k^f = \bigcup_{m \varepsilon M} B_k^m .$$

Let M range over all subsets of E^* and k range over all integers between 1 and n, and after each derived edit is obtained augment E^* by f, (i.e., let $E^* = E^* \cup \{f\}$), and continue. This process will terminate, and when it does let the final E^* be denoted by M_3 and call M_3 the derived edit set. Note that $E \subset M_3$.

Definition: Let E be an explicit edit set and M_3 the set of derived edits. Let M_2 be defined to be the subset of M_3 consisting of edits of the following form:

(a) If $f \varepsilon E$, then $f \varepsilon M_2$.

(b) If $f = \prod_{i=1}^{n} B_i^f$ is a derived edit with contributing edits in the set M and with generating field k, and if $B_k^m \neq A_k$ for all edits m in M, then $f \varepsilon M_2$ if $B_k^f = A_k$. Such an edit f is called an essentially new derived edit.

Remark: According to the definition in Section A, if $f = \prod_{i=1}^{n} B_i^f$ and $g = \prod_{i=1}^{n} B_i^g$ are normal edits, f contains g if $B_i^g \subset B_i^f$ for all i=1,...,n, and f properly contains g if $B_i^g \subset B_i^f$ for all i=1,...,n and $B_i^g \neq B_i^f$ for some i=1,...,n. Also, if f is a derived edit we say that f is a maximal derived edit if f is properly contained in no other derived edit.

Definition: If E is an explicit edit set, we define M_1 to be the maximal edits of M_2. The set M_1 is what Fellegi-Holt defines to be the complete set of edits.

Returning to Example 1 above, by using field F_2 as the generating field, we can generate the implied edit, e_3:

	Field F_1		Field F_2		Field F_3
e_3:	$\{1\}$	x	A_2	x	$\{2\}$,

also written as:

edit e_3

$\{0-14\}$ and $\{$ spouse $\}$.

This new edit makes explicit a prohibited response combination involving only fields F_1 and F_3. For this example, the set $\{ e_1, e_2, e_3 \}$ forms the complete set of edits for the explicit edit set $\{ e_1, e_2 \}$.

Example 2: The following is a somewhat more lengthy example and one which we will return to later. This example is found in [GA] and the fields are considered only as discrete sets. Let the range of fields F_i for i=1,...,6 be:

$A_1 = \{ 1,2 \}$ $A_4 = \{ 1,2,3,4 \}$
$A_2 = \{ 1,2,3 \}$ $A_5 = \{ 1,2,3 \}$
$A_3 = \{ 1,2 \}$ $A_6 = \{ 1,2,3,4 \}$.

The explicit edits are:

Field Edit	F_1		F_2		F_3		F_4		F_5		F_6
e_1:	A_1	x	$\{1,2\}$	x	$\{1\}$	x	A_4	x	$\{1,2\}$	x	A_6
e_2:	$\{2\}$	x	A_2	x	$\{2\}$	x	$\{1,2\}$	x	A_5	x	$\{3,4\}$
e_3:	$\{1\}$	x	$\{2,3\}$	x	A_3	x	$\{2,3,4\}$	x	A_5	x	A_6
e_4:	A_1	x	$\{1,3\}$	x	A_3	x	A_4	x	A_5	x	$\{1,2\}$
e_5:	$\{2\}$	x	A_2	x	A_3	x	$\{1\}$	x	$\{2,3\}$	x	A_6

The complete set of derived edits consists of those edits listed below augmented by the explicit edit set $\{e_1, e_2, e_3, e_4, e_5\}$.

Field / Edit	F_1		F_2		F_3		F_4		F_5		F_6
e_6:	A_1	x	$\{2,3\}$	x	$\{2\}$	x	$\{2\}$	x	A_5	x	$\{3,4\}$
e_7:	A_1	x	A_2	x	$\{1\}$	x	A_4	x	$\{1,2\}$	x	$\{1,2\}$
e_8:	A_1	x	$\{2\}$	x	A_3	x	$\{2\}$	x	$\{1,2\}$	x	$\{3,4\}$
e_9:	A_1	x	$\{3\}$	x	$\{2\}$	x	$\{2\}$	x	A_5	x	A_6
e_{10}:	$\{1\}$	x	A_2	x	$\{1\}$	x	$\{2,3,4\}$	x	$\{1,2\}$	x	A_6
e_{11}:	$\{1\}$	x	A_2	x	A_3	x	$\{2,3,4\}$	x	A_5	x	$\{1,2\}$
e_{12}:	$\{2\}$	x	A_2	x	$\{1\}$	x	$\{1\}$	x	A_5	x	$\{1,2\}$
e_{13}:	$\{2\}$	x	$\{1,2\}$	x	A_3	x	$\{1,2\}$	x	$\{1,2\}$	x	$\{3,4\}$
e_{14}:	$\{2\}$	x	$\{1,2\}$	x	A_3	x	$\{1\}$	x	A_5	x	$\{3,4\}$
e_{15}:	$\{2\}$	x	$\{1\}$	x	A_3	x	$\{1\}$	x	A_5	x	A_6
e_{16}:	$\{2\}$	x	$\{1\}$	x	A_3	x	$\{1,2\}$	x	$\{1,2\}$	x	A_6
e_{17}:	$\{2\}$	x	$\{1,2\}$	x	$\{1\}$	x	$\{1\}$	x	A_5	x	A_6
e_{18}:	$\{2\}$	x	$\{1,3\}$	x	$\{2\}$	x	$\{1,2\}$	x	A_5	x	A_6.

In general, given a derived edit it is difficult to determine which explicit or previously derived edits were employed in its derivation. We note in passing that edits e_2 and e_3 using generating field F_1 combined to imply edit e_6, and edits e_5 and e_{13} using generating field F_5 combined to imply edit e_{14}.

Remark: (F-H) If f is a derived edit, there exists g ε M_1 (i.e., a maximal derived edit) such that g contains f.

Remark: If a response vector fails an edit in any one of E, M_1, M_2, or M_3, it fails an edit in each of E, M_1, M_2, and M_3. We also observe that $M_1 \subset M_2 \subset M_3$.

C. Continuous Data

In this section we assume A_i equals the set of non-negative real numbers, A, for all $i=1,\dots,n$. That is, $\underline{a} = (a_1,\dots,a_n)$ is an n-tuple of non-negative reals.

Definition: A linear inequality edit, e, is the region in $\mathbf{A}^n$ defined by an inequality of the type:

$$e: \sum_{i=1}^{n} f_i x_i > b.$$

If $f_k \neq 0$, we say field k enters edit e.

Definition: An edit set having M edits, H, is a collection of edits:

$$H = \{ e_j : \sum_{i=1}^{n} f_{ij} x_i > b_j \mid j=1,\ldots,M \}.$$

Remark: Succumbing to a slight abuse of terminology, we will usually refer to the linear inequality

$$\sum_{i=1}^{n} f_i x_i > b$$

as an edit (as opposed solely to the region it determines). Thus, the edit set H is really a family of subsets of $\mathbf{A}^n$ and the region determined by all the edits in H is the union of these subsets of $\mathbf{A}^n$.

Definition: The feasible region determined by an edit set H, denoted by T, is defined to be:

$$T = \{ \underline{x} = (x_1,\ldots,x_n) \in A^n \mid \sum_{i=1}^{n} f_{ij} x_i \leq b_j \text{ for all } j=1,\ldots,M \}.$$

Note that the feasible region is the intersection of a family of "half-planes" (really "half-hyperplanes") and hence is a convex region, and in fact, a convex polyhedron in n-dimensional space. Conforming to the definitions in Section A, an edit set H is consistent if T is not empty. A record $\underline{a} \in \mathbf{A}^n$ is consistent if $\underline{a} \in T$, otherwise $\underline{a}$ is said to be inconsistent or invalid.

Remark: Accordingly, if $\underline{a} \in \prod_{i=1}^{n} \mathbf{A}^n$ is a record and $e: \sum_{i=1}^{n} f_i x_i > b$ is an edit, we say that:

(i) a fails e if: $\sum_{i=1}^{n} f_i a_i > b$, and

(ii) a passes e if: $\sum_{i=1}^{n} f_i a_i \leq b$.

Thus, a record $\underline{a} \in \mathbf{A}^n$ is said to be consistent if it passes all edits; which is equivalent to failing no edits.

Example 3: The following is a simple example of a continuous editing scenario. Suppose we have only three fields, $A_1 = A_2 = A_3 = \mathbf{A}$, and we have the following explicit linear inequality edits:

$$\begin{array}{lll} e_1: & -x_1 + 2x_2 & > 0 \\ e_2: & x_1 - 4x_2 & > 0 \\ e_3: & -2x_2 + x_3 & > 0 \\ e_4: & x_2 - x_3 & > 0 . \end{array}$$

Suppose also that we have the following three records:

$$\underline{r}_1 = (800,300,400)$$
$$\underline{r}_2 = (800,300,200)$$
$$\underline{r}_3 = (400,300,900) .$$

Note that record $\underline{r}_1$ fails no edits and hence is consistent. Record $\underline{r}_2$ fails edit e_4 and record $\underline{r}_3$ fails edits e_1 and e_3; hence both of these records are inconsistent and are considered invalid, (neither lies in the feasible region defined by edits e_1 through e_4).

Remark: According to the definition in Section A, if

$$f: \sum_{i=1}^{n} f_i x_i > b$$

$$g: \sum_{i=1}^{n} g_i x_i > c$$

are two linear inequality edits, then f contains (respectively, properly contains) g if the region determined by f contains (respectively, properly contains) the region determined by g. If the domain of each field were **R**, all reals, rather than all non-negative reals, the region determined by

$$f: \sum_{i=1}^{n} f_i x_i > b$$

contains the region determined by

$$g: \sum_{i=1}^{n} g_i x_i > c$$

if and only if there exists an $k > 0$ such that:

$$g_i = kf_i \qquad \text{for all } i=1,\ldots,n \text{ and}$$

$$c > kb .$$

In many applications, the domain for each field is the non-negative reals. In such cases there are the implied constraints $x_i \geq 0$ for all $i=1,\ldots,n$, and more care must be exercised in determining whether one edit dominates another.

Definition: Let

$$e_1: \sum_{i=1}^{n} f_{i1} x_i > b_1$$

$$e_2: \sum_{i=1}^{n} f_{i2} x_i > b_2$$

be two edits, let k be an integer $1 \leq k \leq n$, and suppose f_{k1} and f_{k2} are non-zero and have opposite signs. Letting g_{k1} and g_{k2} be the absolute values of f_{k1} and f_{k2} respectively,

$$e_3: \sum_{i=1}^{n} (g_{k2} f_{i1} + f_{i2} g_{k1}) x_i > g_{k2} b_1 + g_{k1} b_2$$

is an edit. Note that the coefficient of x_k in e_3 is zero, and we define e_3 as the essentially new edit derived from e_1 and e_2 with generating field k.

Remark: As in the categorical case, we can define the set of derived edits based on a family of explicit edits E. Let $E^* = E$ be an explicit edit set. Consider all pairs of elements in E^* such that the coefficients of x_k have opposite sign and let k range over the integers $\{1, \ldots, n\}$. Form an essentially new derived edit, h, as indicated above, augment E^* by h, (i.e., let $E^* = E^* \cup \{h\}$), and continue this process. The set of implied edits that can be derived in this fashion is referred to as the essentially new derived edit set (as in the categorical case). In fact, all one really cares about are the maximal essentially new derived edits.

Example 4: In this example we will use edits having only two fields, and the explicit edit set will contain three edits. The explicit edit set consists of:

$$e_1: -x_1 + 2x_2 > 10$$

$$e_2: x_1 + x_2 > 10$$

$$e_3: 2x_1 - x_2 > 10 .$$

The derived edits are:

$$e_4: x_2 > 20/3$$

$$e_5: x_2 > 10$$

$$e_6: x_1 > 10$$

$$e_7: x_1 > 20/3 .$$

We obtained: e_4 from e_1 and e_2 with generating field 1,
e_5 from e_1 and e_3 with generating field 1,
e_6 from e_1 and e_3 with generating field 2,
e_7 from e_2 and e_3 with generating field 2.

Note that edit e_5 is contained in edit e_4 and that e_6 is contained in edit e_7. The set of maximal essentially new derived edits is $\{e_1, e_2, e_3, e_4, e_7\}$.

In general, if f and g are two edits and field k enters both edits with the opposite sign, then the derived edit using generating field k will form a hyperplane parallel to the k-axis. This is illustrated in **Figure 1** for Example 4. The edit-failing region for each of the explicit edits, $\{ e_1, e_2, e_3 \}$, lies in the direction of the arrow away from the corresponding line. That is, the line labeled e_1 is the line

$$-x_1 + 2x_2 = 10.$$

The arrow directed above that line is the region

$$-x_1 + 2x_2 > 10.$$

Similar considerations hold for each of the implied edits. In this figure, the shaded area is the feasible region (i.e., the region T discussed above). The derived edit e_4 corresponds to the area above the broken line through (10/3, 20/3) parallel to the x_1 - axis, and the edit e_5 corresponds to the area above the line through (10,10) parallel to the x_1-axis. Clearly, the edit failing region determined by e_5 is contained in that determined by e_4, and we can see that e_5 is not a maximal edit. Similar considerations apply to edits e_6 and e_7 (not drawn) and one can see that e_6 is not maximal by considering the inequalities above.

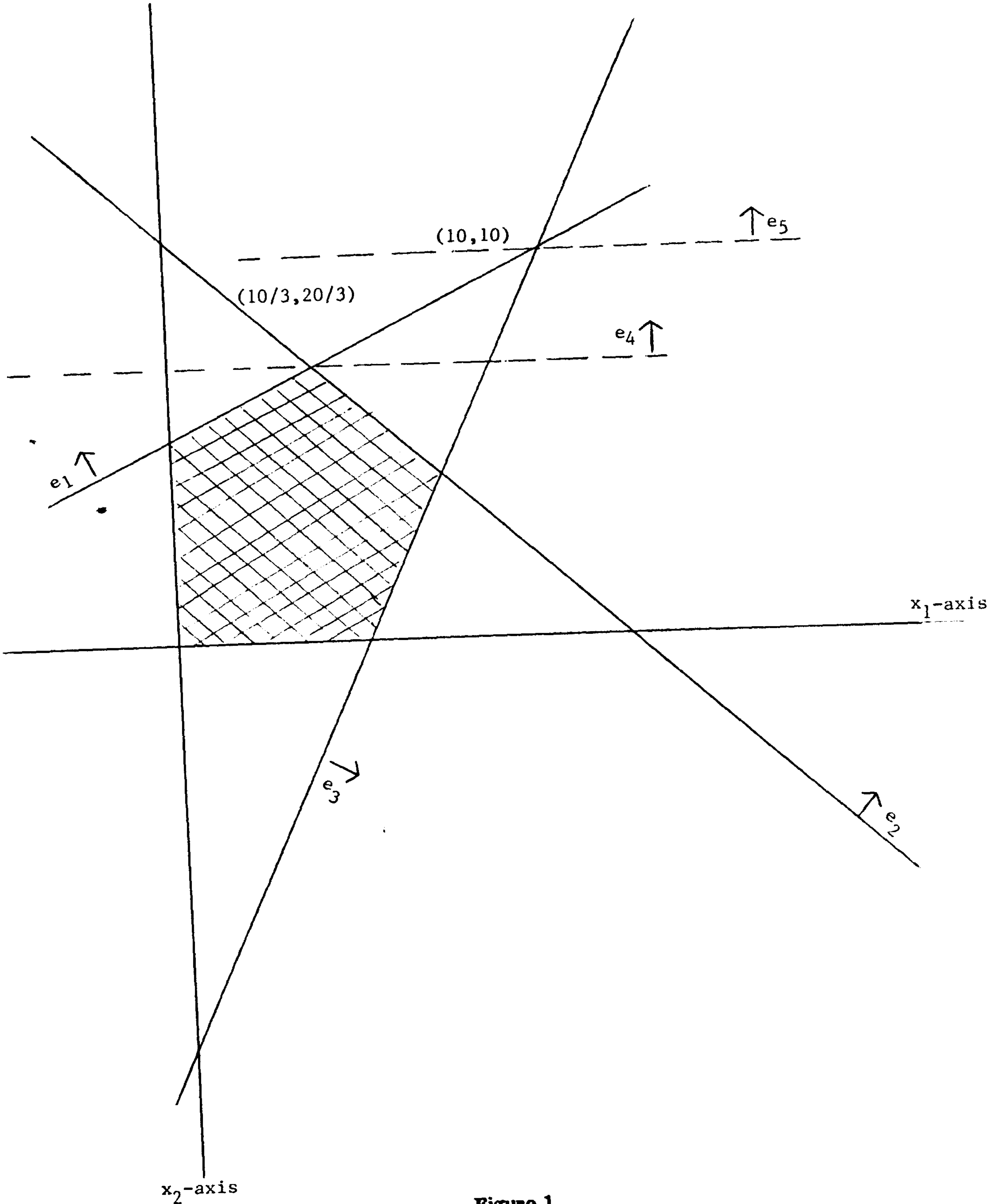

Figure 1.

III. CHARACTERIZING MINIMAL DELETION SETS

In Chapter II we discussed detecting the presence of an inconsistent record (with respect to an explicit edit set) by observing whether any explicit edits are failed by the record under consideration. In general, determining that a record is not consistent does not suffice for most applications of editing. One would like to know which fields can be changed on an inconsistent record so that the record can be made consistent. The obverse question is to ask which fields on a record are themselves mutually consistent. Of course, if one could determine which fields it suffices to change on an edit-failing record to create a valid record, the remaining fields must be mutually consistent; and conversely.

In order to answer these related questions one must employ edits derived from the (user supplied) explicit edit set. In the previous chapter, we showed how to derive implied edits from an explicit edit set and gave examples, both for categorical and continuous data. In this chapter we formalize the relation between (1) fields to delete (on an edit-failing record) (2) a mutually consistent subset of fields, and (3) the complete set of edits.

Definition: Let $S \subset \{1, \ldots, n\}$ and $\underline{a} = (a_1, \ldots, a_n)$. If there exists a consistent record $\underline{b} = (b_1, \ldots, b_n)$ such that $a_i = b_i$ for all $i \in S$ we say that the set of response variables $\{a_i\}_{i \in S}$ is a consistent set of variables for $\underline{a}$. If $\{a_i\}_{i \in S}$ is a consistent set of variables on a record $\underline{a}$, we say that the set $\{a_i\}_{i \notin S}$ is a deletion set for a.

Remark: Note that in this definition we relate a consistent set of variables on a record and those variables to be changed so that the entire record can be made consistent. A deletion set consists of exactly those variables on the record that it suffices to change so that the entire record can be made consistent. The remaining variables on a record are viewed as (mutually) consistent.

It might appear that we could have defined a consistent subset of fields on a record to be those fields mutually failing no explicit edit. But this is not quite right as the following examples show.

Example 5: Returning to Example 1, consider the record

$$\underline{r}_3 = (12, \text{Widowed, Spouse}).$$

If we delete the response "Widowed" (namily field F_2) from this record, we observe that the remaining fields F_1 and F_3 fail neither of the explicit edits e_1 or e_2. However, there is no possible response to the field "Marital Status" that can consistently complete this record if at least one of the fields F_1 and F_3 is not changed. The difficulty is that the responses "12 years old" and "Spouse" are not (mutually) consistent. By generating edit e_3, we see that the combination "12 years old" and "Spouse" fails this new edit. This constraint was implied by edits e_1 and e_2 but did not surface until edit e_3 was generated.

Example 6: Returning to Example 2, consider the record

$$\underline{r} = (2,1,1,1,2,1) \quad .$$

This record fails the explicit edits e_1, e_4, and e_5. Suppose we were to delete fields F_2 and F_5. Note that the remaining four fields fail none of the five explicit edits. Even so, the responses 2,1,1,1 on fields F_1, F_3, F_4 and F_6 respectively are not mutually consistent. In fact, they fail implied edits: e_{12}, e_{15}, e_{16}, and e_{17}. We note in passing that the field values 1,2,1 on fields F_4, F_5 and F_6 are mutually consistent according to the definition above. In fact if we let $F_1 = 1$ $F_2 = 2$ and $F_3 = 2$, and allow F_4, F_5, and F_6 to remain as 1,2, and 1, respectively, then we have the following consistent record

$$(1,2,2,1,2,1) \; .$$

Remark: In the two preceeding examples we used phraseology stating that a set of fields on some record fails no edit or does fail some edit. In Chapter II, we defined what it means for a record to fail an edit, but made no corresponding definition for a subset of the field values on a record. We hoped a reader could sense the meaning in the context above and we now provide a precise defintion.

Definition: Let $\underline{a} = (a_1,\ldots,a_n) = \prod_{i=1}^{n} A_i$ be a record and let $\{a_i\}_{i \varepsilon S}$ be a set of field values on $\underline{a}$. We say that the set $\{a_i\}_{i \varepsilon S}$ fails edit, e, if:

(i) (for categorical data)

$$a_i \varepsilon \; B_i \text{ for all } i \varepsilon S \text{ and } A_i = B_i \text{ for all } i \notin S \text{ where } e = \prod_{i=1}^{n} B_i \; ,$$

(ii) (for continuous data)

$$\sum_{i=1}^{n} f_i a_i > c \text{ and } f_i = 0 \text{ for all } i \notin S \text{ where } e: \sum_{i=1}^{n} f_i x_i > c.$$

Remark: Let $\underline{a} = (a_1,\ldots,a_n)$ be a record, H a subset of the complete set of edits and $Q \subset \{1,\ldots,n\}$, then

(i) if $\{a_i\}_{i \notin Q}$ fails an edit, $e \in H$, then $\underline{a}$ also fails e,

(ii) if Q is a deletion set for $\underline{a}$, then $\{a_i\}_{i \notin Q}$ fails no edits in H.

Proof: (i) Observing that the only entering fields of e are contained in $\{F_i \mid i \notin Q\}$, the result follows.

(ii) Since Q is a deletion set there exists a consistent record, $\underline{b} = (b_1,\ldots,b_n)$, such that $b_i = a_i$ for $i \notin Q$. If an edit, $e \in H$, fails $\{a_i\}_{i \notin Q}$, then e also fails $\{b_i\}_{i \notin Q}$, so e also fails $\underline{b}$ by (i). This is a contradiction since $\underline{b}$ was assumed consistent.

Definition: Let $\underline{a} = (a_1,\ldots,a_n) \in \prod_{i=1}^{n} A_i$ be a response vector and let H be an arbitrary subset of the complete set of edits. Let $H_{\underline{a}}$ be the set consisting of all edits in H failed by $\underline{a}$, and denote a typical element of $H_{\underline{a}}$ by e_h. Let Q be a subset of $\{1,\ldots,n\}$ with the property that for each $e_h \in H_{\underline{a}}$ there exists a $t \in Q$ such that field F_t enters edit e_h. We say that the set of fields, Q, is a cover of the failed edit set $H_{\underline{a}}$.

Example 7: We return (once again) to Example 1 and record

$$\underline{r}_3 = (12, \text{Widowed, Spouse}) = (1,4,2).$$

When considering the complete set of derived edits, $H = \{e_1, e_2, e_3\}$, we note that record $\underline{r}_3$ fails each of these edits, so $H_{\underline{r}_3} = H$. Edit e_1 has entering fields F_1 and F_2, edit e_2 has entering fields F_2 and F_3 and edit e_3 has entering fields F_1 and F_3. If we let $Q = \{1,2\}$ we see that Q is a cover for $H_{\underline{r}_3}$ since for each edit either field F_1 or F_2 enters (or both).

Example 8: Returning to Example 6, let

$$\underline{r} = (2,1,1,1,2,1).$$

If we let $H = \{e_i \mid i=1,...,18\}$ be the complete set of edits, we can observe that the edit set that fails record $\underline{r}$ is the set:

$$H_{\underline{r}} = \{e_1,\ e_4,\ e_5,\ e_7,\ e_{12},\ e_{15},\ e_{16},\ e_{17}\}\ .$$

Note that edit:

e_1 has entering fields: $F_2\ F_3\ F_5$,

e_4 has entering fields: $F_2\ F_6$,

e_5 has entering fields: $F_1\ F_4\ F_5$,

e_7 has entering fields: $F_3\ F_5\ F_6$,

e_{12} has entering fields: $F_1\ F_3\ F_4\ F_6$,

e_{15} has entering fields: $F_1\ F_2\ F_4$,

e_{16} has entering fields: $F_1\ F_2\ F_4\ F_5$,

e_{17} has entering fields: $F_1\ F_2\ F_3\ F_4$.

If we let $Q = \{1,2,3\}$ we see that each failed edit has at least one of F_1, F_2 or F_3 as an entering field. Thus, Q is a cover of $H_{\underline{r}}$.

Instead of letting H in this example consist of the complete set of derived edits, let us see what happens if we let $H = \{e_1,\ e_2,\ e_3,\ e_4,\ e_5\}$ be the set of explicit edits. In this case, the edits failed by $\underline{r}$ form the set $H_{\underline{r}} = \{e_1,\ e_4,\ e_5\}$. We observe that $Q = \{1,2\}$ is a cover of $H_{\underline{r}}$. However, the fields F_1 and F_2 are certainly not a deletion set for $\underline{r}$. For, if we let F_3, F_4, F_5, and F_6 remain as 1,1,2 and 1, respectively; no values of F_1 and F_2 could complete this record to form a complete consistent record.

The contrast between letting H be the set of explicit edits rather than the complete set of derived edits is crucial. As noted above, when we considered only the explicit edits, the set of fields $\{ F_1, F_2 \}$ formed a cover of $H_{\underline{r}}$ but they were not a deletion set for

$$\underline{r} = (2,1,1,1,2,1)$$

because the field values r_3, r_4, r_5, r_6 are not mutually consistent. In contrast, the cover $Q = \{ 1,2,3 \}$ of $H_{\underline{r}}$ where H is complete set of edits does yield a deletion set for $\underline{r}$. That is, the values r_4, r_5, r_6 are mutually consistent. The result we one leading to is as follows: if $\underline{r}$ is a record and H is the complete set of edits, then a cover of $H_{\underline{r}}$ is a deletion set for $\underline{r}$.

Remark: Given a record $\underline{a}$ and a subset, H, of the complete set of edits, the task of finding a cover for $H_{\underline{a}}$ can be simplified considerably by viewing the problem in terms of a zero-one matrix. For the record $\underline{a}$ we will define the failed edit matrix, $M_{\underline{a}}$. The rows will be indexed by the edits failed by $\underline{a}$ and the columns will be indexed by all fields $\{ F_i \mid i=1,\ldots,n \}$. We define the entries of $M_{\underline{a}}$ by:

$$M_{\underline{a}}(e,i) = \begin{cases} 1 & \text{if } F_i \text{ enters edit } e \\ 0 & \text{otherwise.} \end{cases}$$

Thus, if H equals the complete set of edits from Example 8, and $\underline{r} = (2,1,1,1,2,1)$, the failed edit matrix $M_{\underline{r}}$, is:

	F_1	F_2	F_3	F_4	F_5	F_6
(e_1)	0	1	1	0	1	0
(e_4)	0	1	0	0	0	1
(e_5)	1	0	0	1	1	0
(e_7)	0	0	1	0	1	1
(e_{12})	1	0	1	1	0	1
(e_{15})	1	1	0	1	0	0
(e_{16})	1	1	0	1	1	0
(e_{17})	1	1	1	1	0	0

We seek a family of columns, Q, of this matrix such that each row has at least one non-zero entry in one of the columns in Q. If we let Q = { 1, 2, 3 } we see that each row has at least one "1" in the first three columns. If we had let Q = { 4,5,6 } the same would be true. The columns in Q correspond to fields forming a cover for the set of failed edits $H_{\underline{a}}$.

Remark: It should be clear by now, that it does not suffice to use only the explicit set of edits to determine fields to delete on a record and to find a consistent subset of field values. On the other hand, it does suffice to use the complete set of derived edits. We will now elaborate on this theme.

Definition: We say a subset H of the general derived edit set (denoted in Chapter II by M_3) is sufficient if for every $\underline{a} = (a_1,\ldots,a_n) \in \prod_{i=1}^{n} A_i$, every cover of $H_{\underline{a}}$ is a deletion set.

Remark: It is clear that if H is a sufficient edit set and $H \subset L$, then L is also sufficient.

Remark: The crowning result of the Fellegi-Holt paper [FH] is that the complete set of derived edits (denoted by M_1 in Chapter II) is a sufficient set of edits. Thus, if $\underline{a}$ = $(a_1,\ldots,a_n)$ is a record and H is the complete set of edits, a cover for $H_{\underline{a}}$ will be a deletion set for $\underline{a}$. For a proof of this result we refer the interested reader to [FH]. Thus, if Q is a cover for $H_{\underline{a}}$, then the field values $\{a_i\}_{i \in Q}$ form a deletion set and so the field values $\{a_i\}_{i \notin Q}$ are consistent.

Theorem: [FH] The complete set of edits is a sufficient set of edits.

Proposition: Let $\underline{a} = (a_1,\ldots,a_n)$ be a record, H an arbitrary subset of the complete set of edits, and $Q \subset \{1,\ldots,n\}$. If Q is a deletion set for $\underline{a}$, then Q is a cover of $H_{\underline{a}}$.

Proof: Since Q is a deletion set for $\underline{a}$, the set of values, $\{a_i\}_{i \notin Q}$, is consistent and hence every edit failed by $\underline{a}$ must have a least one entering field in $\{ F_i \mid i \in Q \}$. Thus, Q is a cover of $H_{\underline{a}}$.

Corollary: Let $\underline{a} = (a_1,\ldots,a_n)$ be a record, H be the complete set of edits, and $Q \subset \{1,\ldots,n\}$. The following are equivalent:

(i) $\{a_i\}_{i \notin Q}$ is a consistent set of field values,

(ii) $\{a_i\}_{i \in Q}$ is a deletion set for $\underline{a}$,

(iii) $\{a_i\}_{i \notin Q}$ fails no edits in H,

(iv) Q is a cover of $H_{\underline{a}}$.

Definition: For each field on a questionnaire response record, F_i for $i=1,\ldots,n$, we can define a field weight, w_i for $i=1,\ldots,n$, to be a positive real number. If S is a set of fields, we can define the weight of S to be:

$$W_S = \sum_{i \in S} w_i \quad .$$

In particular, if $\underline{a} = (a_1,\ldots,a_n)$ is an edit failing record, and if Q is a deletion set for $\underline{a}$, we define the weight of Q to be

$$W_Q = \sum_{i \in Q} w_i \quad .$$

Remark: If Q is a cover of $H_{\underline{a}}$, we say Q is a minimum cover if Q properly contains no other cover of $H_{\underline{a}}$. Since all weights are assumed positive, every cover of minimum weight is also a minimum cover. If we select the weight of each field to be equal to 1, the weight of a set of fields is equal to the number of elements in that set.

A common and useful way to assign weights is to let them play the role of preference factors. In so doing, one gives higher weights to the more reliable fields. Thus, given an edit-failing record for which more than one set of fields could serve as a deletion set, one selects the set of fields to delete having the minimal total weight.

In Example 8, we considered the record:

$$\underline{r} = (2,1,1,1,2,1),$$

and observed that either $Q = \{1,2,3\}$ or $Q' = \{2,3,4\}$ are deletion sets for $\underline{r}$. That is, deletion sets are certainly not unique. By computing the weights W_Q and $W_{Q'}$, one would usually select the deletion set of minimal weight to delete fields on an edit-failing record.

Example 9: If we consider the record

$$\underline{r}_2 = (72, \text{Widowed, Spouse})$$

in Example 1, we see that either $Q = \{2\}$ or $Q' = \{3\}$ can serve as a deletion set for this edit-failing record. If it were felt that "marital status" were (in general) a more reliable field than "relation to head," one might have assigned weights to be: $w_2 = 3$ and $w_3 = 2$. Thus the weight of set Q' is less than that of Q so one would delete "Spouse" from the response record. A new value (either "head" or "other") would then be imputed at a later stage of processing.

Example 10: The purpose of this example is to show how this process plays out for a simple case of continuous edits. Let us return to the explicit edits of Example 3:

$$\begin{aligned}
e_1&: \quad -x_1 + 2x_2 > 0\\
e_2&: \quad x_1 - 4x_2 > 0\\
e_3&: \quad -2x_2 + x_3 > 0\\
e_4&: \quad x_2 - x_3 > 0.
\end{aligned}$$

When we add the following two derived edits we obtain the sufficient set of edits:

$$\begin{aligned}
e_5&: \quad -x_1 + x_3 > 0\\
e_6&: \quad x_1 - 4x_3 > 0.
\end{aligned}$$

If we consider the record:

$$r = (800, 500, 300)$$

we obtain the failed edit matrix

$$\begin{array}{c} \\ e_1 \\ e_4 \end{array}
\begin{array}{c} \begin{array}{ccc} F_1 & F_2 & F_3 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \end{array}.$$

The field F_2 is a cover for the failed edits e_1 and e_4, hence field F_2 is a deletion set for (800, 500, 300). When we leave the values $x_1 = 800$ and $x_3 = 300$, we find the feasible region for x_2 is determined by the constraints:

$$\begin{aligned} -800 + 2x_2 &\leq 0 \\ 800 - 4x_2 &\leq 0 \\ -2x_2 + 300 &\leq 0 \\ x_2 - 300 &\leq 0 . \end{aligned}$$

Thus, the record (800, x_2, 300) will be consistent if and only if

$$200 \leq x_2 \leq 300 .$$

A somewhat more complex example follows from the record:

(500, 300, 1000).

The failed edit matrix is:

$$\begin{array}{c} \\ e_1 \\ e_3 \\ e_5 \end{array} \begin{array}{c} \begin{array}{ccc} F_1 & F_2 & F_3 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \end{array} .$$

We must choose two fields to change, and choosing x_2 and x_3 and leaving $x_1 = 500$ we have that (500, x_2, x_3) is consistent if and only if x_2 and x_3 satisfy all constraints:

$$\begin{aligned} x_2 &\leq 250 \\ x_2 &\geq 125 \\ -2x_2 + x_3 &\leq 0 \\ x_2 - x_3 &\leq 0 \\ x_3 &\leq 500 \\ x_3 &\geq 125 . \end{aligned}$$

The set of points (x_2, x_3) for which (500, x_2, x_3) satisfies all edits e_1 through e_6 lies in the shaded region of **Figure 2.**

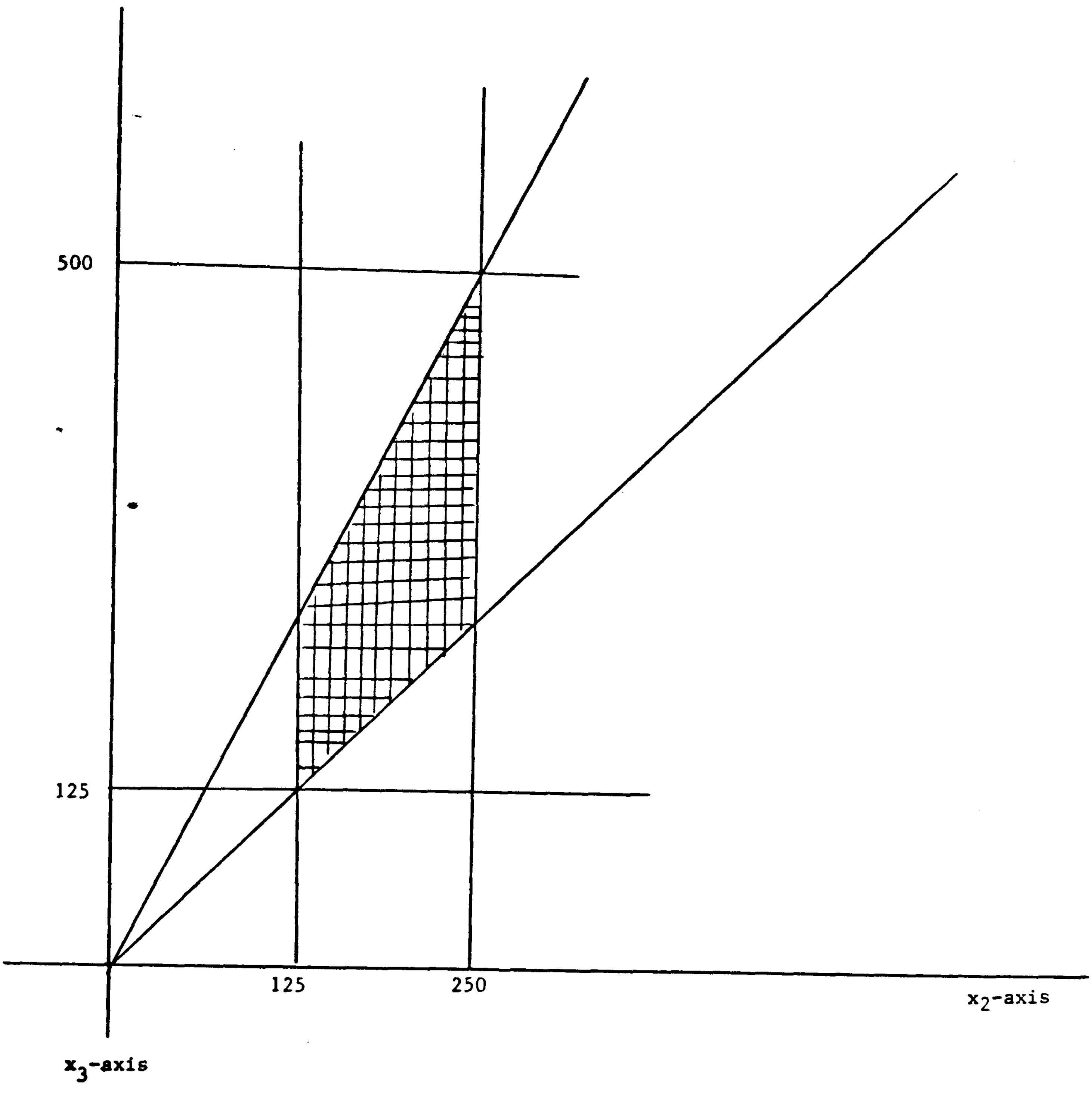

Figure 2

Remark: Note that implied edits do not provide any new information as to which records are consistent or inconsistent for completely reported records. That is, a completely reported record which passes all explicit edits will also pass all implied edits. If however, a record contains some nonresponse, the derived edits may be crucial in determining if the reported fields are mutually consistent. If we consider the edits in Example 1, and the record:

Age = 8

Marital Status = Missing

Relation to Head = Spouse,

we observe that neither explicit edit, e_1 or e_2, is failed. The record does fail implied edit e_3, and it is clear that one of the reported fields must be changed.

What this all says is the following. Given a record $\underline{a}$ which fails some edits (explicit or derived), we would like to locate a subset of fields and change only those fields so that the revised record becomes consistent. The set of fields that are changed is called the deletion set and the fields not changed are (mutually) consistent. As a rule the objective is to find a minimum deletion set (and hence, a maximum consistent set). But more generally, one assigns weights to each field and attempts to locate a weighted minimal deletion set; that is, subset of variables whose sum of weights is minimal. This problem is often referred to as the minimum (weighted) fields to delete problem.

Given a record $\underline{a}$ and the set of failed edits $H_{\underline{a}}$, where H is the complete set derived edits, one finds a (minimal weighted) deletion set for $\underline{a}$ by finding a (minimal weighted) cover for $H_{\underline{a}}$. The problem of finding a minimal weighted cover for $H_{\underline{a}}$ is a fairly standard integer programming problem in operations research called the set covering problem. We will discuss the set covering problem in the next section and show how it is used in finding a minimal weighted set of fields to delete on an edit failing record. For a detailed discussion of the minimal weighted fields to delete problem from an operations research perspective we refer the reader to [GKL] and [LGK]. An alternative to the set covering procedures to locate a minimal weighted set of fields to delete on edit-failing records is discussed in [S]. The methods developed there are based on mathematical programming procedures.

IV. USING A SET COVERING PROCEDURE TO FIND A MINIMAL DELETION SET

In the proceeding sections, given a sufficient set of edits, H, and an edit failing record, a, we saw that it suffices to find a cover Q of $H_{\underline{a}}$ in order to identify a set of fields to alter on a in order to create a consistent record. In order to find the cover of $H_{\underline{a}}$, one in effect, must solve a set covering problem. In Section A, we give a precise formulation of the set covering problem and in Section B we relate it to editing.

A. The Set Covering Problem

Definition: Let $G = \{g_i \mid i \in 1,...,m\}$ be an arbitrary finite set and let $P = \{P_j \mid j=1,...,n\}$ be a family of subsets of G. We say $C \subset P$ is a cover for G if

$$G = \bigcup_{P_j \in C} P_j .$$

If we associate a weight $w_j > 0$ with each $P_j \in P$, we can define the weight of a cover C to be

$$W_C = \sum_{P_j \in C} w_j .$$

One says that a cover is a minimum cover if it properly contains no other cover. Note that since all weights are positive, a cover of minimum weight is also a minimum cover.

The Set Covering Problem: Given a set $G = \{g_i \mid i=1,...,m\}$, a family of subsets of G, $P = \{P_j \mid j=1,...,n\}$, and a set of positive weight, $W = \{w_i \mid i=1,...,n\}$, one seeks to find a cover of G by P of minimum weight. This is known as the set covering problem.

Example 11: Let G be the set

$$G = \{ 1,2,3,4,5,6,7,8,9,10 \},$$

and consider the set of subsets of G, $P = \{ P_j \mid j=1,...,6 \}$

where

$$P_1 = \{ 1,2,3,4,5,6,7 \}$$

$$P_2 = \{ 5,6,7,8,9,10 \}$$

$$P_3 = \{ 1,3,5,7,9 \}$$

$$P_4 = \{ 2,4,6,8,10 \}$$

$$P_5 = \{ 3,8 \}$$

$$P_6 = \{ 1,2,4 \} .$$

Using the elements of P, we can find the covers of G; some are:

$$C_1 = \{ P_1, P_2 \}$$

$$C_2 = \{ P_3, P_4 \}$$

$$C_3 = \{ P_2, P_5, P_6 \}$$

$$C_4 = \{ P_2, P_3, P_6 \}$$

$$C_5 = \{ P_1, P_3, P_4 \} .$$

In each case, the union of the sets in a cover equal the entire set $G = \{ g_i \mid i=1,\ldots,10 \}$. In this example, C_5 is <u>not</u> a minimum cover since it contains C_2, however all other covers listed are minimum.

<u>Remark</u>: We can form the matrix M whose rows are indexed by the elements of G and whose columns are indexed by P, and

$$M(i,j) = \begin{cases} 1 & \text{if } g_i \in P_j \\ 0 & \text{otherwise.} \end{cases}$$

In the case of Examle 11, the matrix M is:

	P_1	P_2	P_3	P_4	P_5	P_6
g_1	1	0	1	0	0	1
g_2	1	0	0	1	0	1
g_3	1	0	1	0	1	1
g_4	1	0	0	1	0	1
g_5	1	1	1	0	0	0
g_6	1	1	0	1	0	0
g_7	1	1	1	0	0	0
g_8	0	1	0	1	1	0
g_9	0	1	1	0	0	0
g_{10}	0	1	0	1	0	0

In general, the rows in M correspond to elements in G, and column j is thought of as corresponding to set P_j where $g_i \in P_j$ if the element in the i^{th} row and j^{th} column of M is equal to 1.

By selecting a family of columns such that each row contains at least one non-zero entry in one of the specified columns, the set corresponding to the selected columns forms a cover for G. For example, by choosing columns P_3 and P_4 we see that each row has a "1" in either column P_3 or P_4. Thus, $\{P_3, P_4\} = C_2$ is a cover for G.

Reformulating the Set Covering Problem: The Set Covering Problem can be formulated as follows. Given:

(a) a set $G = \{g_i \mid i=1,\ldots,m\}$
(b) a family of subsets of G, $P = \{P_j \mid j=1,\ldots,n\}$
(c) a set of positive weights $W = \{w_j \mid j=1,\ldots,n\}$,

Minimize $$W = \sum_{j=1}^{n} w_j x_j$$

Subject to $$\sum_{j=1}^{n} a_{ij} x_j \geq 1 \qquad i=1,\ldots,m ,$$

where $x_j \in \{0,1\}$ $j=1,\ldots,n$,

and $$a_{ij} = \begin{cases} 1 \text{ if } g_i \in P_j \\ 0 \text{ otherwise.} \end{cases}$$

A cover of minimum weight is:

$$C = \{ P_j \mid x_j = 1 \text{ for } j=1,\ldots,n \}.$$

B. Applying the Set Covering Problem to Find a Minimal Deletion Set: When we apply the set covering problem in the context of data editing, we let H be the complete set of edits, $G = \{ h_1,\ldots,h_m \} = H_{\underline{a}}$ be the edits failing record $\underline{a}$, P_j the set of edits which field F_j enters for $j=1,\ldots,n$, and w_j the weight of field F_j for $j=1,\ldots,n$. Thus, the corresponding matrix, $M_{\underline{a}}$, has rows indexed by the edits in $H_{\underline{a}}$, columns indexed by the fields, and

$$M_{\underline{a}}(i,j) = \begin{cases} 1 \text{ if field } j \text{ enter failed edit } i \\ 0 \text{ otherwise.} \end{cases}$$

If we set up the matrix $M_{\underline{a}}$ for the record and edits in Example 8, we get the exact matrix on Page 19.

Since the sets P_j correspond to the set of failed edits which field j enters, if C is a cover in the sense above, then

$$H_{\underline{a}} = \bigcup_{P_j \in C} P_j .$$

Of course, this was our objective all along. To be more explicit, the set of fields,

$$Q = \{ F_j \mid P_j \in C \text{ for } j=1,\ldots,n \}$$

is a cover of $H_{\underline{a}}$ in the sense of Chapter II. Thus, the set of fields,

$$Q = \{ F_j \mid P_j \in C \text{ for } j=1,\ldots,n \}$$

is a deletion set for $\underline{a} = (a_1,\ldots,a_n)$, and the set of field values

$$\{ a_j \mid P_j \notin C \text{ for } j=1,\ldots,n \}$$

is consistent.

<u>Summary</u>: If $\underline{a} = (a_1,\ldots,a_n)$ is an edit-failing record and H is the complete set of edits (for an explicit edit set) to find a minimal (weighted) deletion set for $\underline{a}$, we solve the following <u>zero-one integer programming</u> problem.

$$\text{Minimize} \sum_{j=1}^{n} w_j x_j$$

$$\text{subject to } M_{\underline{a}} \geq 1$$

where $\underline{x} = (x_1,\ldots,x_n)$ is a vector of zeros and ones,

$$H_{\underline{a}} = \{ h_i \mid i=1,\ldots,m \},$$

$$M_{\underline{a}}(i,j) = \begin{cases} 1 & \text{if field } F_j \text{ enters edit } h_i \\ 0 & \text{otherwise, and} \end{cases}$$

$\underline{w} = (w_1,\ldots,w_n)$ is a vector of positive field weights.

A minimal weighted deletion set for $\underline{a}$ corresponds to the fields F_j such that $x_j = 1$. The field values a_j such that $x_j = 0$ are mutually consistent and need not be altered. That is, we need only change the field values a_j for $x_j = 1$ to obtain a consistent record.

<u>Remark</u>: To solve the problem above, one, in essence, solves a set covering problem. In addition to exact procedures, efficient heuristic techniques that approximate optimal solutions can yield acceptable results. In the next two sections we discuss two programs that implement the methods discussed in this report.

V. PROGRAMS IMPLEMENTING THE METHODOLOGY DISCUSSED ABOVE

Several programs are in place at the Census Bureau to implement edit generation procedures for explicit edit sets and set covering techniques for edit-failing records.

One set of programs handles categorical data and a second set handles continuous data under ratio edits. They are both discussed below.

A. Implementing the Set Covering Procedures for Categorical Data

We have several programs at the Census Bureau to implement the set covering procedures discussed earlier for categorical data. These programs are based on software developed at Oak Ridge National Laboratory, and this software is discussed in [L]. The first program, call it GENED, generates a sufficient set of implied edits from a user-supplied explicit edit set.

In the Appendix we include the output of GENED when this program was run on Example 1 and Example 3 above. This program itself is divided into two segments. In the first segment, the system prompts the user for the number of fields, field names, number of responses for each field, and the possible response options. The program then "feeds back" to the user this information to be verified or changed. This segment of the program is illustrated on page A1 of the Appendix for Example 1. Next the user is requested to supply the explicit edit set. For the edits in Example 1, the user supplied edits are shown on page A2 of the Appendix. The program then generates the implied edits, and these are shown on page A3 of the Appendix.

The program GENED terminates and the implied edits are stored in a file. These implied edits can now be read into a second program to edit individual data records. If the purpose of running GENED at this stage was not to immediately edit records but rather to analyze the user-supplied explicit edits, the derived edits are available to do so. Through an examination of the logical implications of the explicit edit set (conveyed by the implied edits), a user may wish to modify the original explicit edits. Even if a user does not wish to edit records using the set covering approach, the information provided by the implied edits can be quite valuable in evaluating an explicit edit set and its logical implications.

The GENED program was also run on the explicit edit set from Example 2, and the output is contained in pages A6 through A9 of the Appendix. The implied edits are listed on pages A8 and A9 and they can be easily compared to the implied edits listed in Example 3 on page 8; in fact, this program was the source of these edits.

The second program we have available, called EDRECS, edits records using the implied edits generated earlier. The program first prompts the user to furnish a weight for each field so that the program can select a minimal weighted set of fields to delete for each edit-failing record. On page A4 of the Appendix we include the output of running records $\underline{r}_1$, $\underline{r}_2$, $\underline{r}_3$ from Example 1. The weights were selected to be equal to 1. One sees the input record (in coded form), the list of fields to change, and the weight (or cost) of the minimal deletion set. Since all the weights were selected to be 1, the cost of the solution is the number of elements in the deletion set. On pages A10 and A11 of the Appendix, we show the output from running several records based on Example 2. After a complete "batch" of records is processed through this system, the program displays the frequency with which each edit failed. On page A5 we display this frequency count for the records from Example 1 and on page A12 we show the results for the records from Example 2. This information is potentially quite useful in an analysis of the impact of the edits on the data processed. In addition, this information may indicate edits that need revision.

B. Implementing the Fellegi-Holt Procedures for Linear Inequality Edits from Ratio Constraints

Linear inequality edits as discussed in Chapter II, Section C can arise from ratio edits, namily the requirement that the ratio of two fields lie between two specified bounds.

That is, a ratio edit between fields F_h and F_k is the requirement that

$$L_{hk} \leq x_h / x_k \leq U_{hk}$$

where L_{hk} and U_{hk} are constants. Each ratio edit gives rise to two linear inequality edits

$$e_1: \quad - x_h + L_{hk} x_k > 0$$

$$e_2: \quad x_h - U_{hk} x_k > 0.$$

Given two ratio edits:

$$L_{hk} \leq x_h / x_k \leq U_{hk}$$

$$L_{kp} \leq x_k/x_p \leq U_{kp},$$

we can derive the implied edit:

$$L_{hk}L_{kp} \leq x_h/x_p \leq U_{hk}U_{kp}.$$

Given a family of connected explicit ratio edits:

$$L_{hk} \leq x_h/x_k \leq U_{hk} \quad ,$$

we can easily obtain all maximal essentially new implied edits and their number is quite manageable, namly n(n-1) where n is the number of fields. We then have a sufficient edit set and can proceed to edit records using the set covering approach. The set covering problem that arises has a particularily simple structure since each edit has exactly two entering fields. Special set covering procedures can be used in this setting and they are discussed in [GR]. In fact, the SPEER System (Structured Program for Economic Editing and Referrals) developed at the Census Bureau has a set covering procedure as its foundation.

The primary purpose of SPEER is to provide an edit and imputation system for economic data under ratio edits. The system is divided into three major segments: (1) edit generation, (2) error localization (determining a weighted minimal set of fields to delete on edit-failing records), and (3) imputation subroutines. The first two segments proceed as discussed in earlier sections. The imputation subroutines consist of a family of structured modules in which subject-matter specialists insert survey-specific imputation rules. Within the imputation segment of this system, we sequentially impute one field at a time. As we do this, the system explicitly generates the one-dimensional feasible region for each field being imputed to ensure that each imputation is consistent with all other fields on the record. Thus, the imputation subroutines sequentially generate a family of mutually consistent field values.

This program has been sucessfully used to process six portions of the 1982 Economic Censuses. For a further discussion of SPEER, we refer the reader to [GS].

REFERENCES

[FH] Fellegi, I.P., and D. Holt. (1976). A Systematic Approach to Automatic Edit and Imputation, JASA, 71, 17-35.

[GA] Garfinkel, R.S. (1979). An Algorithm for Optimal Imputation of Erroneous Data. College of Business Administration Working Paper Series. The University of Tennessee, Knoxville.

[GKL] Garfinkel, R.S., Kunnathur, A.S., and Liepins, G.E., Optimal Imputation of Erroneous Data: Categorical Data, General Edits. (To Appear, Operations Research).

[GN] Garfinkel, R.S. and G.L. Nemhauser (1972). Integer Programming. Wiley. New York.

[GR] Greenberg, B. (1981). Developing an Edit System for Industry Statistics. Computer Science and Statistics: Proceedings of the 13th Symposium of the Interface. 11-16, Springer-Verlag, New York.

[GS] Greenberg, B., Surdi, R. (1984). A Flexible and Interactive Edit and Imputation System. Proceedings of the Section on Survey Research Methods. American Statistical Association.

[L] Liepins, G.E. (1984). Algorithms for Error Localization of Discrete Data. Oak Ridge National Laboratory.

[LGK] Liepins, G.E., Garfinkel, R.S., and Kunnathur, A.S. (1982) Error Localization for Erroneous Data: A Survey, TIMS Studies in the Management Sciences, 19, 205-219.

[S] Sande, G. (1979) Numerical Edit and Imputation. International Association for Statistical Computing. 42nd Session of the International Statistics Institute.

APPENDIX

Here we include sample output produced by the programs discussed in the text for the implementation of the set covering procedures for automated editing of categorical data. The contents of this Appendix are discussed in Section A of Chapter V.

field #	field name	code	code value
1	age		
		1	0-14
		2	15+
2	marital status		
		1	single
		2	married
		3	divorced
		4	widowed
		5	separated
3	rel to head		
		1	head
		2	spouse
		3	other

edit number	entering fields	
edit # 1	age 0-14	marital status married divorced widowed separated
edit # 2	marital status single divorced widowed	rel to head spouse

the execution has been completed

edit number	entering fields	
edit # 1	age 0-14	marital status married divorced widowed separated
edit # 2	marital status single divorced widowed	rel to head spouse
edit # 3	age 0-14	rel to head spouse

the execution has been completed

```
*** THE WEIGHTS IN FIELD ORDER ARE ***
1.0000  1.0000  1.0000


enter response codes as a vector of integers
after last record enter a vector of integers
with first entry = -10

for record #     1
THE INPUT RECORD IS:
 2  4  1
THE INPUT RECORD IS PASSING

for record #     2
THE INPUT RECORD IS:
 2  4  2
THE FIELDS TO BE CHANGED ARE:
  3
THE WEIGHT OF THE SOLUTION IS:     1.0000

for record #     3
THE INPUT RECORD IS:
 1  4  2
THE FIELDS TO BE CHANGED ARE:
  3   1
THE WEIGHT OF THE SOLUTION IS:     2.0000
```

edit #	# of times involved in failure
1	1
2	2
3	1

A5

field #	field name	code	code value
1	field a		
		1	res a1
		2	res a2
2	field b		
		1	res b1
		2	res b2
		3	res b3
3	field c		
		1	res c1
		2	res c2
4	field d		
		1	res d1
		2	res d2
		3	res d3
		4	res d4
5	field e		
		1	res e1
		2	res e2
		3	res e3
6	field f		
		1	res f1
		2	res f2
		3	res f3
		4	res f4

edit number	entering fields			
edit # 1	field b res b1 res b2	field c res c1	field e res e1 res e2	
edit # 2	field a res a2	field c res c2	field d res d1 res d2	field f res f3 res f4
edit # 3	field a res a1	field b res b2 res b3	field d res d2 res d3 res d4	
edit # 4	field b res b1 res b3	field f res f1 res f2		
edit # 5	field a res a2	field d res d1	field e res e2 res e3	

the execution has been completed

edit number	entering fields				
edit # 1	field b res b1 res b2	field c res c1	field e res e1 res e2		
edit # 2	field a res a2	field c res c2	field d res d1 res d2	field f res f3 res f4	
edit # 3	field a res a1	field b res b2 res b3	field d res d2 res d3 res d4		
edit # 4	field b res b1 res b3	field f res f1 res f2			
edit # 5	field a res a2	field d res d1	field e res e2 res e3		
edit # 6	field b res b2 res b3	field c res c2	field d res d2	field f res f3 res f4	
edit # 7	field c res c1	field e res e1 res e2	field f res f1 res f2		
edit # 8	field b res b2	field d res d2	field e res e1 res e2	field f res f3 res f4	
edit # 9	field b res b3	field c res c2	field d res d2		
edit #10	field a res a1	field c res c1	field d res d2 res d3 res d4	field e res e1 res e2	
edit #11	field a res a1	field d res d2 res d3 res d4	field f res f1 res f2		
edit #12	field a res a2	field c res c1	field d res d1	field f res f1 res f2	
edit #13	field a res a2	field b res b1 res b2	field d res d1 res d2	field e res e1 res e2	field f res f3 res f4
edit #14	field a res a2	field b res b1 res b2	field d res d1	field f res f3 res f4	
edit #15	field a res a2	field b res b1	field d res d1		
edit #16	field a res a2	field b res b1	field d res d1 res d2	field e res e1 res e2	
edit #17	field a res a2	field b res b1	field c res c1	field d res d1	

```
                                          res b2
edit #18           field a           field b            field c            field d
                       res a2            res b1             res c2             res d1
                                         res b3                                res d2
```

A9

the execution has been completed

```
*** THE WEIGHTS IN FIELD ORDER ARE ***
1.0000  1.0000  1.0000  1.0000  1.0000  1.0000


enter response codes as a vector of integers
after last record enter a vector of integers
with first entry = -10

for record #     1
THE INPUT RECORD IS:
 2  1  1  1  2  1
THE FIELDS TO BE CHANGED ARE:
  5   6   2
THE WEIGHT OF THE SOLUTION IS:     3.0000

for record #     2
THE INPUT RECORD IS:
 1  2  2  2  1  2
THE FIELDS TO BE CHANGED ARE:
  4
THE WEIGHT OF THE SOLUTION IS:     1.0000

for record #     3
THE INPUT RECORD IS:
 1  3  2  4  1  1
THE FIELDS TO BE CHANGED ARE:
  6   2
THE WEIGHT OF THE SOLUTION IS:     2.0000

for record #     4
THE INPUT RECORD IS:
 2  3  1  3  3  4
THE INPUT RECORD IS PASSING

for record #     5
THE INPUT RECORD IS:
 1  2  2  1  1  1
THE INPUT RECORD IS PASSING

for record #     6
```

THE INPUT RECORD IS:

1 1 1 1 1 1

THE FIELDS TO BE CHANGED ARE:

6 2

THE WEIGHT OF THE SOLUTION IS: 2.0000

for record # 7

THE INPUT RECORD IS:

2 2 2 2 2 2

THE INPUT RECORD IS PASSING

for record # 8

THE INPUT RECORD IS:

1 3 2 4 3 4

THE FIELDS TO BE CHANGED ARE:

4

THE WEIGHT OF THE SOLUTION IS: 1.0000

edit #	# of times involved in failure
1	2
2	0
3	3
4	3
5	1
6	0
7	2
8	0
9	0
10	0
11	2
12	1
13	0
14	0
15	1
16	1
17	1
18	0