MATCHING OBSERVATIONS® ON BOTH CONTINUOUS AND CATEGORICAL VARIABLES: A SIMPLE APPLICATION OF THE FASTCLUS PROCEDURE

Manuel S. Lombardero, University of Pittsburgh, Pittsburgh, PA

Abstract

Data analysts in the area of biomedical research often need to match observations from two different data sets according to one or more factors. The data handling capabilities of BASE SAS® provide the analyst with flexible tools that facilitate the matching of observations, even when this matching process is complicated by the presence of a large number of matching factors, or by the inclusion of continuous variables among the matching factors. However, when perfect matches cannot be attained, observations may need to be matched on the basis of how close they are, and not just on whether they are identical. This report presents a SAS macro that finds perfect matches when they exist, and otherwise it finds the closest matches available. This macro relies primarily on the FASTCLUS procedure which regards observations as points in a multi-dimensional space. The use of the macro is illustrated by means of examples, and its strengths and weaknesses are discussed.

Introduction

The need to match observations according to their values in one or more variables arises frequently in the analysis of data in health-related fields. For example, investigators may wish to compare the clinical outcome of patients who receive a new treatment to those who receive the standard or no treatment. In case-control studies, epidemiologists routinely need to identify subjects who are not afflicted by some medical condition (e.g., leukemia), in order to compare their odds of exposure (to some agent such as radiation) to that of another group of individuals who suffer from the condition. Whether the study is prospective or retrospective, the investigator must make sure that any observed association (or lack thereof) between exposure and outcome is not due to the spurious effect of some confounding variable. The effect of potential confounders can be eliminated with stratification and regression methods, or, alternatively, by ensuring that the two groups being compared are similar with regards to those variables. The latter approach has two distinct advantages. One, it is conceptually simple and it can be easily communicated to audiences not familiar with regression methodologies. Two, it facilitates the comparative analysis of a large number of diverse features (continuous, categorical, or combinations) by freeing the analyst from the burden of having to identify appropriate regression techniques for each of these. These two advantages may sometimes compensate for some of the disadvantages of the matched analysis approach, such as the loss of observations (and the corresponding reduction of statistical power), and the logistical difficulties in obtaining appropriate matches.

The task of matching observations from two groups can range from trivial to very complex. The basic data handling capabilities of BASE SAS (the DATA step, PROC SORT, PROC FREQ, etc.) suffice to match observations on a few categorical variables. If (some of) the matching variables are continuous, they can be categorized and handled similarly. This will, however, often entail the introduction of arbitrary divisions. For example, grouping patients into the three age groups < 45, 45-64, ≥ 65, implies that matching a 45-year old to a 64-year old is acceptable, but matching a 64-year old to a 65-year old is not. A better solution for continuous variables, but one which still involves some arbitrariness, is to allow a certain difference in the value of the matching variable (such as, say, 5 years in our age example) between the observations being matched. This type of matching specification can be handled with some effort with the traditional tools in BASE SAS, or in a relatively straightforward way with PROC SQL, as it has recently been suggested (Corelle, 1993).

The current report introduces a SAS macro that assists the analyst in the process of matching observations when perfect matches are not feasible. This can easily happen when the number of matching variables is large, and/or when some or all of the matching variables are continuous. This macro, which will attempt to find the closest possible matches, is particularly useful in settings where the investigator is willing to tolerate some differences between the two observations being matched. The macro has been named FCMATCH. Its first two letters come from the SAS/STAT® procedure FASTCLUS, the workhorse of the macro.

The analysis that motivated the development of FCMATCH involved liver transplant recipients from a multi-center liver transplantation registry (NIDDK Liver Transplantation Database) housed in the Epidemiology Data Center, University of Pittsburgh. Investigators had to match suitable controls to liver transplant recipients who had undergone a procedure (TIPS) for the treatment of gastro-intestinal bleeding (N=53). These controls had to be chosen from all the patients available in the registry at the time (N=579). Investigators wanted to compare the peri-operative clinical outcome of treated patients to that of a group of untreated patients that would otherwise resemble the first group. This resemblance was defined in terms of nine variables which offered a relevant profile of the clinical (diagnosis), demographic (age), and circumstantial (transplantation center) characteristics of each patient. A primitive version of FCMATCH was quickly created and used to carry out this analysis. Since then, it has been enhanced and tested in other data sets.

General Description

For the remainder of this report, the discussion of the general problem of matching observations will take place in the context of the following specific setting. One group of observations will be thought of as representing patients or subjects with some characteristic of interest (they may have received some treatment, or they may suffer from some medical condition). These observations will be referred to as *cases*. A larger group of observations will be thought of as representing patients or subjects who do not have that particular characteristic and, as a result, are potentially useful as controls for the cases. The observations from this larger group will be referred to as *candidates*. The problem at hand is to select $n \geq 1$ candidates for each case on the basis of their similarity with regards to

characteristics specified by a set of variables, referred to as the *matching factors*, or *MFs*. Selected candidates will be interchangeably referred to as either *controls* or *matches* (depending of the context) for the appropriate case.

The first step toward matching appropriate controls to cases is to visualize both cases and candidates as points in a Euclidean space whose dimensions correspond to the MFs. With this arrangement in mind, it becomes clear that one way to identify suitable controls is to be able to detect "clusters" of candidates surrounding, i.e., being close to, each case. A brief survey into the capabilities for clustering available in the SAS System revealed the FASTCLUS procedure (SAS/STAT User's Guide, Version 6, Fourth Edition) was the ideal tool for our purpose. FASTCLUS is primarily used for disjoint cluster analyses on large (up to 100,000 observations) data sets, and its capabilities extend well beyond those required by FCMATCH. By selecting the appropriate options, one can force FASTCLUS to treat the cases as the centers of clusters. Then FASTCLUS assigns each candidate to the closest case. For each case, FCMATCH chooses its n (specified by the user) closest candidates and reports them to the user.

Examples

The examples presented here illustrate the use of FCMATCH, including calls to the macro, interpretation of its output, and some additional aspects of its inner workings which are relevant to its application. All three examples are based on simulated data sets with continuous matching variables. The first example is a straight application of the macro, in its default settings. The second and third examples are intended to describe an apparent anomaly in the behavior of the macro when applied with the default settings, and a simple corrective measure which is an integral part of FCMATCH.

Example 1

A data set SET1 with 10,000 observations is created. Of these, 10 are cases, and the rest are candidates. There are 5 MFs (X1,X2,...,X5), and all of them take integer values from 0 to 100. Ten of the candidates are created to have each MF value identical to those of the cases. Thus, the macro should be able to identify at least one perfect match for each case. The remaining candidates are assigned MF integer values randomly. The data set includes an indicator variable TMT with a value of 1 denoting cases and a value of 0 for candidates. The variable I ranges in value from 1 to 10,000, and will be used as an observation identifier. The following call to FCMATCH requests that 5 controls be identified for each case:

```
%FCMATCH (
        data=set1,
        caseind=tmt,
        matchby=x1 x2 x3 x4 x5,
        id=i,
        out=set1m,
        n=5
        );
```

Meaning of parameters: **data** contains the name of the SAS data set with the observations; **caseInd** contains the name of the indicator variable that identifies cases; **matchby** contains a text string listing the MFs; **Id** contains the name of some appropriate observation identifier; **out** contains the name of the data set that FCMATCH creates (see more below); and **n** contains the number of controls requested for each case.

The final product of FCMATCH is a SAS data set with all the necessary matching information. By default, FCMATCH prints the content of this data set. An excerpt is shown in Output Frame 1. All the cases for which at least one match could be found, and all the matches found, are represented in this data set, with each observation corresponding to either a case or a match. In this example, matches for 10 cases were requested. If a complete set of matches (5) had been found for each, the output data set would have a total of $(10 \times 5) = 50$ controls. In this run, however, FCMATCH only found 47 of them. The excerpt shown reveals the case (I=10) for which FCMATCH was unable to find a complete set of controls.

Output Frame 1

MATCH	I	TMT	DISTANCE	TI
1	10	1	5.5453	10
	20	0	0.0000	10
	7245	0	11.0905	10
2	6	1	6.8940	6
	16	0	0.0000	6
	4930	0	6.1644	6
	5062	0	8.8882	6
	7566	0	9.2195	6
	5083	0	10.1980	6
3	2	1	8.0180	2
	12	0	0.0000	2
	2451	0	5.7446	2
	3461	0	7.6811	2
	9281	0	11.2694	2
	3058	0	15.3948	2
⋮	⋮	⋮	⋮	⋮
10	1	1	14.2170	1
	11	0	0.0000	1
	7337	0	11.9164	1
	9912	0	15.1327	1
	4295	0	17.6352	1
	5022	0	26.4008	1

The observations in this output data set are grouped by *matching groups*, where a matching group consists of one case and its matches. The variable MATCH identifies the matching group. The observation identifier and the treatment indicator (in this example, I and TMT, respectively) are carried along from the input data set. An additional identifier is also created. It is named by concatenating the name of the observation identifier to the letter 'T' (producing, in this example, the name 'TI'). In a given matching group, the value of this additional identifier is the same value of the main observation identifier that corresponds to the case in that matching group. This added variable makes it easy for the user to merge controls with their associated cases from some other data set, if this need should arise. The variable DISTANCE reports the separation between the case and each of its controls. If the observation is a case, then DISTANCE represents the average distance between that case and all the controls that were assigned to it. Notice that the matching groups are ordered according to this mean distance value, with the groups having the shortest mean distance positioned on

top. All the perfect matches that were included in the input data set were identified; in fact, if perfect matches exist, FCMATCH will always find them. In the four matching groups displayed above, these perfect matches can be easily recognized because their associated DISTANCE value is zero.

FCMATCH also reports, whenever applicable, the cases with incomplete matches:

Output Frame 2

Cases with incomplete number of controls

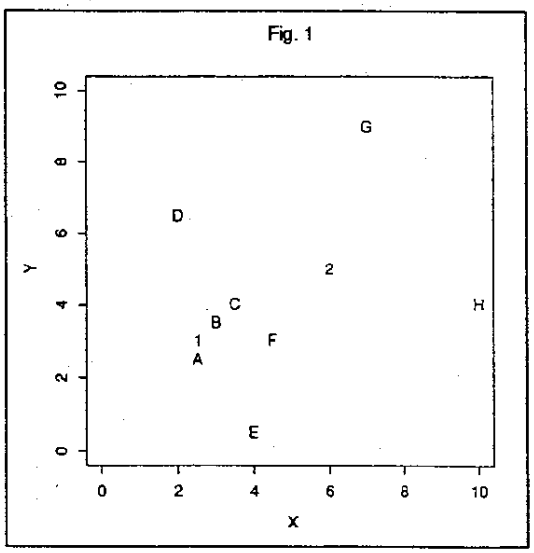
OBS	CASE ID	NO. OF CONTROLS FOUND
1	10	2

Once the analyst is satisfied with the matches obtained, this data set can be re-sorted by the observation identifier (**Id**=I), and merged with the original data set (**data**=SET1) to resume the analysis. □

Example 2

The data set in this example (SET2) is so small (2 cases, 8 candidates) that, in practice, any matching would be more easily done by hand. Its small size will, however, allow us to illustrate some important operational features of FCMATCH. As before, the variable TMT is used to identify cases. The main observation identifier this time is a character variable called TAG. It is used to "tag" cases with integers ("1" and "2"), and candidates with letters ("A" through "H"). There are only two MFs, X and Y, so the observations can be plotted in a two-dimensional plane according to their (X,Y) value (see Fig. 1).

Fig. 1



In this example the aim will be to obtain 2 controls for each case. It is clear by simple inspection that "A" and "B" should be assigned to "1", and "C" and "F" should be assigned to "2". A call to FCMATCH in this case would be:

```
%FCMATCH (
        data=set2,
        caseind=tmt,
        matchby=x y,
        id=tag,
        out=set2m,
        n=2
        );
```

The output data set SET2M looks as follows:

Output Frame 3

MATCH	TAG	TMT	DISTANCE	TTAG
1	1	1	0.60355	1
	A	0	0.50000	1
	B	0	0.70711	1
2	2	1	4.12311	2
	G	0	4.12311	2
	H	0	4.12311	2

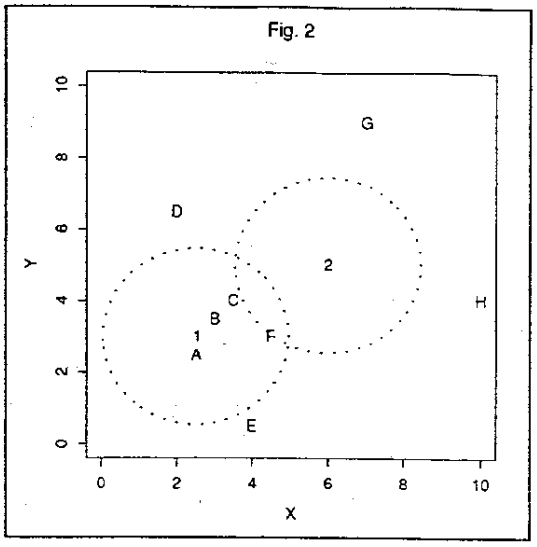
The matches found for case "1" are the ones to be expected, but the matches assigned to case "2" are clearly not the best ones. What happened? Very simply, FASTCLUS assigns each candidate to the closest case, and both "C" and "F" are closer to "1" than they are to "2". Therefore, they are assigned to case "1" (regardless of whether "1" needs them or not), and become unavailable to all other cases.

FCMATCH can circumvent this behavior by including a parameter which requires the macro to invoke FASTCLUS several times. An iterative process is set up where matches that satisfy a decreasingly stringent case-control distance requirement are chosen as acceptable matches for each iteration. Specifically, during the i^{th} iteration in a run with p iterations, a candidate will be accepted as a permanent match for some case only if the separation between the two does not exceed some radius R_i, which is set to equal the $100\times(i/p)^{th}$ percentile (as determined by PROC UNIVARIATE with its default settings) of all the case-candidate distances reported by FASTCLUS during that iteration. These cases (and their associated candidates) are removed from the input data set, which is then left with the unmatched cases. The removal of the best matched cases frees up some candidates, which become available in the next iteration to cases that are more peripherally located.

Thus, to improve the matches the user specifies that the matching process be accomplished in more than one step or iteration. FCMATCH is called a second time, with the request that two steps be carried out (the default number of steps is one):

```
%FCMATCH (
        data=set2,
        caseind=tmt,
        matchby=x y,
        id=tag,
        out=set2m,
        n=2,
        steps=2
             );
```

Fig. 2

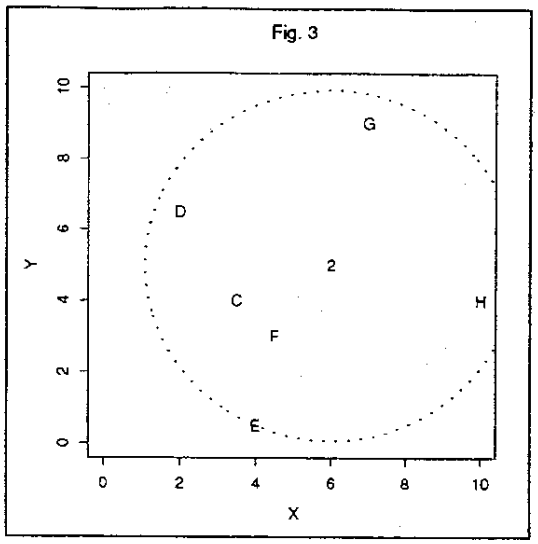


In the first step, only cases that are within a radius of 2.4577 (in this case, this is the 100(1/2)th = 50th percentile of all the case-candidate distances that exist at the beginning, i.e., during the first iteration) units of their matches will be accepted. As illustrated in Fig. 2, case "2" (with tentative matches "G" and "H") fails to satisfy this requirement.

FCMATCH removes case "1", along with its matches "A" and "B", and repeats the process with the remaining observations (plotted in Fig. 3). Notice that in the second and last iteration, "C" and "F" are available to case "2". They are also within R_2=4.9244 units of case "2", as are, in fact, all other candidates (by design, the case-control distance requirement during the last iteration is essentially non-existent, since R_{last} will always be the largest case-candidate distance). The output data set from this two-step run is presented in Output Frame 4.

In addition to the printout of its output data set, FCMATCH also reports summary statistics of the case-control distances. These are displayed below (Output Frame 5),

Fig. 3



for both the 1-step and 2-step runs just described.

Output Frame 4

MATCH	TAG	TMT	DISTANCE	TTAG
1	1	1	0.60355	1
	A	0	0.50000	1
	B	0	0.70711	1
2	2	1	2.59629	2
	C	0	2.69258	2
	F	0	2.50000	2

Notice that with the 2-step run FCMATCH achieved a reduction in the mean case-control distance of about 32%. In most real-life problems, this reduction is less dramatic. □

There are no strict guidelines for deciding how many steps are required for a given problem. Rarely have any substantial gains been found beyond 5-7 steps. In general, however, users should

Output Frame 5

Matching statistics for a run with 1 Step(s)

Steps In Which Matches Were Added	Number Of Matches Added	Mean Case-control Distance	Minimum Case-Control Distance	Maximum Case-Control Distance	Maximum Case-Control Distance Allowed	Clock Time Elapsed In This Run
1	4	2.36333	0.5	4.12311	4.1231	
Total	4	2.36333	0.5	4.12311		0:00:15

Matching statistics for a run with 2 Step(s)

Steps In Which Matches Were Added	Number Of Matches Added	Mean Case-Control Distance	Minimum Case-Control Distance	Maximum Case-Control Distance	Maximum Case-Control Distance Allowed	Clock Time Elapsed In This Run
1	2	0.60355	0.5	0.70711	2.4577	
2	2	2.59629	2.5	2.69258	4.9244	
Total	4	1.59992	0.5	2.69258		0:00:26

carry out as many runs as they need to feel comfortable that no additional gains are likely beyond a certain number of steps. This kind of exploratory process is illustrated in the next example.

Example 3

The data set used in this example (SET3) has 10,000 observations, 500 of which are cases. TMT is the case indicator, I is the observation identifier, and X1, X2, X3, and X4 are the MFs. Their values are random digits between 0 and 100. The task at hand is to identify two matches per case. FCMATCH is run ten times, increasing the number of steps used in each run from 1 step to 10 steps:

```
%FCMATCH (
        data=set3,
        caseind=tmt,
        matchby=x1 x2 x3 x4,
        id=i,
        out=set3m,
        n=2,
        steps=1,
        printout=no
        );
%FCMATCH (
        data=set3,
        caseind=tmt,
        matchby=x1 x2 x3 x4,
        id=i,
        out=set3m,
        n=2,
        steps=2
        printout=no
        );
        .
        .
        .
%FCMATCH (
        data=set3
        caseind=tmt,
        matchby=x1 x2 x3 x4,
        id=i
        out=set3m
        n=2,
        steps=10
        printout=no
        );
```

The parameter **printout** controls whether the output data set is printed or not. In this case it is suppressed because the closeness of the matches attained in the various runs is the result of interest, rather than the specific matches identified in each run. The matching statistics generated by these runs are summarized in Table 1. The run with 8 steps produced the best set of matches. Not only do dramatic reductions in the mean case-control distance appear unlikely for additional steps (> 10), but the differences on the mean distances between the runs already tried are fairly small relative to the magnitude of the distances themselves. □

The general impression gained from these and other examples is that increasing the number of steps from 1 to 3 or 4 is almost always worthwhile, in particular due to the moderate amount of time involved (~ 2 minutes) even for fairly large data sets (10,000 observations). On the other hand, after 7 or 8 steps the matching improvement tends to be minimal or non-existent. These remarks are based on very limited testing.

Table 1

Number of Steps	Mean case-control distance	Clock Time Elapsed in this run*
1	7.1354**	0:01:26
2	7.1467	0:02:02
3	7.1362	0:02:02
4	7.1283	0:02:03
5	7.1235	0:02:04
6	7.1188	0:02:06
7	7.1174	0:02:07
8	7.1159	0:02:10
9	7.1197	0:02:48
10	7.1181	0:03:09

* Time obtained by SAS 6.08 for OS/2, on a computer with a 486/25 micro processor, and 12Mb of RAM.

** In this run, FCMATCH came up 3 matches short of the total (1000) required.

Additional Features

FCMATCH has some additional features that have not been illustrated by the examples. A brief description of these features is presented here.

Maximum case-control distance specification. FCMATCH allows the user to set a maximum case-control distance with the **maxdis** parameter. This feature can be useful if there are categorical MFs for which agreement is essential. If, say, gender matching has to be perfect, one can code the gender variable MALE as 0 (female) and 100 (male), and then specify a **maxdis** value < 100. **Maxdis** can also be used to set a limit in the number of MFs in which a matching violation will be accepted. For example, if a large number of dichotomous variables are used as MFs, and are all coded as 0 and 1, then the distance between two observations whose values agree on all except x MFs is $\sqrt{x}$. Thus, if differences in no more than, say, 4 MFs should exist between cases and controls, then the value of **maxdis** can be set equal to $\sqrt{4} = 2$.

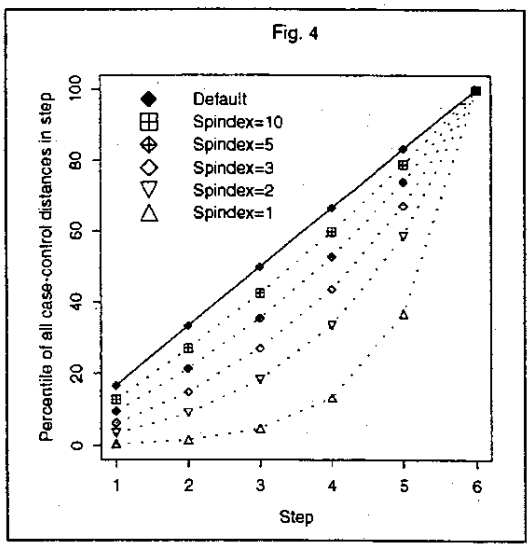
It must be noted that the freedom to code MFs in any arbitrary way is not only a convenient mechanism for excluding unacceptable matches, but, more generally, it is the means by which the analyst can establish matching priorities among the MFs.

Spacing of radii used in the iterations. As Example 2 illustrated, in a run with p > 1 steps or iterations, only candidates located within a certain radius R_i of their nearestcase can be permanently assigned as matches to that case during the i^{th} iteration ($1 \le i \le p$). With its default settings, the value of R_i used by FCMATCH is the $(i/p)^{th} \times 100^{th}$ percentile of all the case-control distances reported by FASTCLUS in the i^{th} iteration. The following formula for determining the percentile for the i^{th} iteration is also available to the user:

$$\text{Percentile} = \frac{e^{i/\text{spindex}} - 1}{e^{p/\text{spindex}} - 1} \times 100\%, \quad \text{spindex} \neq 0$$

where **spindex** (a spacing index) is a user-specified parameter. This produces percentiles that are *not* evenly spaced across the 0-100 interval. As illustrated in Fig. 4, the larger the value of **spindex**, the more the resulting spacing resembles the one obtained by the default setting. The use of positive and small values of **spindex** will result in very small percentile increments during the initial steps followed by large increases toward the final steps. Negative values of **spindex** will have the opposite effect, namely, they will produce large increases initially and small ones at the end (curves corresponding to negative values of **spindex** are not shown in Fig. 4; if drawn, they would be symmetrical to the ones displayed, relative to the default line).

Fig. 4



Cases with same MF value pattern. If two cases share the same values on all MFs, one of them will be ignored by FASTCLUS, which is only concerned with points in space. FCMATCH checks for this condition, and when it finds two or more observations occupying the same point in space, it "shakes them a little" before feeding them to FASTCLUS. The amount of perturbation is random, and about 4 orders of magnitude smaller than the variance of each matching factor (thus, its impact on the choice of matches is negligible).

Concluding Remarks

When should the analyst use FCMATCH? Obviously, this decision should be based on the circumstances of each problem. Specifically, if the existence of sufficient perfect matches is at least plausible, fairly straightforward solutions based on the SQL procedure should be tried first. It should be noted that with PROC SQL one can also identify controls that fail to be an exact match. To apply this solution, however, the user must specify the separation allowed between cases and their matches for each MF. One advantage of the PROC SQL based solution over FCMATCH is that the rules for match acceptability are not restricted to the *symmetric* concept of distance. For example, if age is one of the matching factors, one can require that the age of the case and its control differ by no more than, say, 5 years, and also that the age of the control be *greater than or equal to* that of the case.

If the number of matches that can be identified with direct methods is not sufficient, then FCMATCH has the potential for becoming very valuable, particularly in those situations where imperfect matches are deemed acceptable. The greatest strength of FCMATCH is its ability to use distances in a multi-dimensional space as the basis for identifying close matches. Thus, the separation between potential matches used by FCMATCH is not only based on how the two observations differ on individual continuous variables, but also on the *number* of both categorical and continuous variables in which they differ.

One final note: even though the method implemented by FCMATCH is intuitively reasonable, it remains very informal. Thus, problems that require an optimal matching according to some specific criterion should be handled by algorithms tailored to ensure that the results are optimal in that particular sense.

Acknowledgments

This work has been supported by the National Institute of Digestive and Kidney Diseases (NIDDK), under contract number N01-DK-0-2251. I would like to thank the following individuals (listed alphabetically) for their helpful comments and suggestions: Dr. Steven Belle, Heather Eng, Shannon FitzGerald, Sharon Lawlor, Allan Rosen, Eric Seaberg, and Yuling Wei. Their involvement, whether as testers of the macro, or as reviewers of this report, has been crucial in putting into focus the appropriate role of FSMATCH and in widening the scope of its potential uses.

References

Corelle C., MacLaughlin D., Drew L., and Longacre M. (1993), "Death Matching: An Algorithm and Comparison of its Implementation in PROC SQL vs PROC SQL Plus a DATA Step," Proceedings of the Eighteenth Annual SAS Users Group International Conference, 18, 237-239.

SAS Institute Inc. (1990), SAS/STAT User's Guide, Version 6, Fourth Edition, Cary, NC: SAS Institute Inc.

Author Contact

Manuel S. Lombardero
Epidemiology Data Center
University of Pittsburgh
Parran Hall 127
Pittsburgh, PA 15261

Phone
(412) 624-6170

Fax
(412) 624-3775

e-mail: manuel@vms.cis.pitt.edu