

SOPHISTICATED DATA LINKAGE USING SAS®

Philip J. d'Almada, Battelle Memorial Institute, Atlanta, Georgia

ABSTRACT

A by-product of the very-low-birthweight project at the Centers for Disease Control and Prevention (CDC) was the development of a sophisticated iterative linking program designed and developed in base SAS® software, release 5.18, to link two databases. The success of the program stemmed from the properties of four SAS functions, **INDEX**, **SCAN**, **SUBSTR** and **TRIM**. The master database was a SAS data set of about 3,500 records and stored on mainframe disc. About two-thirds of these records needed to be linked. The transaction database consisted of two years of raw data stored as two files on tapes and each file containing about 100,000 records. The program was applied twice because of the raw files which were therefore linked to annualized subsets of the master data set created by the **SET** and **IF** statements. Within each application of the program, each master data subset was further dichotomized with the **IF** statement according to whether or not a key linking variable had missing values. To actually link a master data subset to a transaction data set at this point, that is, the first iteration, the **MERGE** statement and three or four variables in a **BY** group were used from a four-variable pool, and the data set option, **IN=**, was invoked. Next, matches and nonmatches were output to SAS data sets with **IF...THEN OUTPUT** and **ELSE IF...THEN OUTPUT** statement pairs. Then, matched records were validated by a comparison of validation variables between the linking master data subset and the transaction data set. A comparison used up to three SAS functions, **SCAN**, **SUBSTR** and **TRIM**, nested within a fourth SAS function, **INDEX**, in **IF...THEN OUTPUT** and **ELSE IF...THEN OUTPUT** statement pairs so that validated and nonvalidated records were output to SAS data sets. Finally, nonvalidated records were concatenated with nonmatched records by a **SET** statement and passed to the next iteration that

used another combination of three or four variables in a **BY** group from the same four-variable pool as the first iteration, and the process was repeated. No more than five iterations were used under any of the four conditions of partitioning the master data set for linking. Accepted linkages were interleaved with the **SET** and **BY** statement pair and saved on mainframe disc. The exercise of the properties of these four SAS functions, **INDEX**, **SCAN**, **SUBSTR** and **TRIM**, resulted in a 99.7 per cent acceptance rate for deterministically linking the transaction database to the 2,197 unlinked records of the master data set.

INTRODUCTION

During a study of very-low-birthweight infants at the Centers for Disease Control and Prevention (CDC), an extended data resource was developed by linking records from two databases. The master or analysis database was composed of hospital-abstracted data that overlapped 3 years of births in the State of Georgia and counted about 3,500 records. The transaction or supplemental database comprised two years of the State vital or birth records, and counted about 100,000 births per year. The master database was stored as a SAS data set on disc, and the transaction records were stored as raw data on tapes by birth year. SAS software, release 5.18, was the current software available in an MVS environment on an IBM® mainframe. A deterministic approach was used to link the records whereby only a few variables were involved in the multiple linking processes. Name variables were used for validating matched records.

The first birth-year of records from the master data set, that is, about 1,300 records, had been linked by a predominantly manual process. Consequently, about 2,200 records remained to

be linked. Objectives to be realized while achieving the goal of linking the databases were to (i) develop a more automated system, and (ii) provide more efficient linkages by validating the matched records, and (iii) afford the client a concise and cohesive program source.

METHOD

A schematic of the methodology presented in this section is illustrated in Figure 1. The databases were checked for duplication of ID variables. Variables that were to be used for linking records were examined and some variables from the transaction data file were converted from alphanumeric to numeric while other variables in either data set were recoded. Some variables from the master data set were edited where preliminary examination of these variables showed errors.

Link variables and validation variables from one year of the transaction data file were read into a SAS data set with an **INFILE** and **INPUT** statement pair. The master data set was correspondingly annualized, that is, divided into two subsets, each containing one birth year of data. Linking proceeded on an annualized basis, that is, by linking one master subset at a time. Programmatically, the master data set was dichotomized twice for annualization and for maternal social security number (SSN) being missing or nonmissing by using a **SET** statement and a compound **IF ... AND ...** statement. The transaction data set was merged to the master subset in the first link iteration using the **MERGE** and **BY** statement pair, and invoking the data set option, **IN=**. The conditional statement pair, **IF ... THEN OUTPUT ...** and **ELSE IF ... THEN OUTPUT ...**, was used to generate a data set of matched records and a data set of unmatched records from the master subset. Any records that did not match in this first link attempt were then passed to the next link iteration which differed from the previous iteration in the arrangement of the variables in the **BY** group. The pool of variables used to link records included the maternal SSN, baby's date of birth (DOB), maternal DOB, and the birth hospital.

Consequently, when the maternal SSN was not missing from the master data subset, five link iterations or arrangements of variables were required as follows: (i) maternal SSN-baby's DOB, (ii) maternal SSN, (iii) birth hospital-baby's DOB-maternal DOB, (iv) birth hospital-baby's DOB, and (v) baby's DOB-maternal DOB. When the maternal SSN was missing from the master data subset, no more than four link iterations or arrangements of variables in the **BY** groups were needed. Matched records were validated at each link iteration and the validation process necessarily differed from iteration to iteration because some matches from a previous link iteration could be discarded by the validation process of the current link iteration. Validation processes, however, were similar across link iterations in that maternal and baby's names were the only variables used.

The implementation of these variables was sophisticated because there were two SAS variables used for each of the two names in the transaction data set, and there was only one SAS variable used for each of the two names in the master data subset. Specifically, the SAS data set functions, **SCAN**, **SUBSTR**, and **TRIM**, embedded in the function, **INDEX**, (see Table 1,) were implemented in conditional **IF ... THEN OUTPUT ...** statements, each used in conjunction with a corresponding **ELSE OUTPUT ...** statement.

The three embedded SAS functions were variously applied to the four name variables from the transaction data set and the products therefrom were applied to a crosscheck of one or the other of the name variables from the master data set by the **INDEX** function (see Figure 2). In addition, the **INDEX** function was invoked from two to six times in any one validation process, and the three embedded functions were applied differently for each validation process (see Figure 2). A validated match or true linkage, therefore, was one in which the name variables between the two data sets were acceptably matched. A visual verification of validated matches was effected at each validation process by printing a list using **PROC PRINT**, and the linkages were maintained in a SAS data set.

Table 1. Definitions of SAS data set functions.

INDEX (argument1, argument2): searches the first argument for the character string specified by the second argument

SCAN (argument1, n, delimiter): returns the nth word from the character expression, argument1, where words are separated by delimiter (nonalphabetic characters, including blank, are default)

SUBSTR (argument1, position, n): extracts from argument1, a substring that is n characters long and beginning at the character located at position

TRIM (argument): removes trailing blanks from the character expression, argument

The link or merge variables, baby's DOB, birth hospital, and maternal DOB, were not necessarily record-unique in any of the combinations in which they were applied, that is, in any link iteration, and, consequently, overmatching did occur. As a result, a subset of IDs was generated from the data set of validated matches for that iteration and used to reduce the complement data set of matched records that failed the validation process. The SAS data set option, **IN=**, was used in conjunction with the SAS special variable, **FIRST.id_variable_name** to generate the reduced data set which was also printed with **PROC PRINT** to ensure that true linkages were not invalidated. The next programmatic step was to concatenate such records with the data set of validated records and to pass the remainder of truly invalid matches, if any, to the next link iteration by concatenating these records with the data set of nonlinked records that was generated at the start of that link iteration. Concatenations were accomplished in SAS data steps using a **SET** statement at every step.

Nonlinked or unmatched records from one iteration were passed to the next link iteration

and this was repeated until all records were successfully linked or no more linkages were possible. The final programmatic step, using a **SET** and **BY** statement pair, was to interleave the data sets of linked records that were generated at the end of every link iteration. The final data set of accepted linkages was saved on mainframe disc and, by invoking the **KEEP** statement, included only the two ID variables from the master and the transaction data sets for security.

The SAS program was modified appropriately to effect the successful linkage of the second birth year of records from the master data set. This final data set of linked IDs was also saved in a mainframe disc file.

RESULTS

The acceptance rate (Table 2) for successful linkages was 99.7 percent for both birth years from the master data set. When maternal SSN was unknown in the master data set, all records were successfully linked. However, when maternal SSN was known in the master data set, more than 99 percent of those records were linked. Most linkages occurred during the first link process: 70 to 90 percent across the four combinations of two birth years and maternal SSN being known or unknown. Seven records out of 2,204 remained unlinked, and all had a recorded maternal SSN in the master data set.

CONCLUSION AND DISCUSSION

The goal of linking two data sets was accomplished when 99.7 percent of hospital-abstracted records were acceptably matched to their vital records and validated. In the process, all objectives were satisfactorily met. That is, the link technique was largely automated and included a programmatic means of reliably validating the linkages. Two SAS programs were developed, one for each birth year of the master data set, and there was minimum of customization for each of the two programs. The difference between the programs was in the

design and implementation of the validation process at each link iteration.

The completeness of linkage when the maternal SSN was unknown may be attributed to the fewness of those records that were to be linked. The simplicity of the features of four SAS data set functions was used to surmount the complexity of the task of deterministically linking two data sets in a more automated fashion.

```

IF INDEX (m_mat-name, TRIM (t_mat-maid)) or
  INDEX (m_mat-name, SUBSTR (t_mat-maid, 1, 6)) or
  INDEX (m_mat-name, SCAN (t_mat-first, 1)) or
  INDEX (m_mat-name, TRIM (t_baby-last)) or
  INDEX (m_baby-name, TRIM (t_baby-last))
THEN OUTPUT ...;

IF INDEX (m_mat-name, SCAN (t_mat-first, 1)) or
  INDEX (m_mat-name, SUBSTR (t_mat-maid, 1, 4)) or
  INDEX (m_baby-name, TRIM (t_baby-last)) or
  INDEX (m_baby-name, SCAN (t_baby-first, 1))
THEN OUTPUT ...;

Legend:
  m_mat-name = maternal-name variable from the master data set
  m_baby-name = baby-name variable from the master data set
  t_mat-maid = maternal-maiden-name variable from the transaction data set
  t_mat-first = maternal-first-name variable from the transaction data set
  t_baby-last = baby-last-name variable from the transaction data set
  t_baby-first = baby-first-name variable from the transaction data set

```

Figure 2. Examples of the use of three SAS data functions embedded in a fourth SAS function.

Table 2. Acceptance rates per link iteration by birth year and maternal SSN status

Link Iteration	Unlinked	Year 1 Linked	Accepted (%)		Unlinked	Year 2 Linked	Accepted (%)	Total Accepted (%)
Maternal SSN known								
1	1690	1503	1,503 (88.93)		393	352	352 (89.57)	
2	187	15	14 (7.49)		41	2	2 (4.88)	
3	173	161	161		39	34	34 (87.18)	
4	12	11	7 (58.33)		5	4	2 (40.00)	
5	5	1	1 (20.00)		3	---	---	
Subtotal			1,686 (99.76)				390 (99.24)	2,076 (99.66)
Maternal SSN unknown								
1	101	89	88 (87.13)		20	15	14 (70.00)	
2	13	13	12 (92.31)		6	5	4 (66.67)	
3	1	1	1 (100.0)		2	1	1 (50.00)	
4	---	---	---		1	1	1 (100.0)	
Subtotal			101 (100.0)				20 (100.0)	121 (100.0)
Total			1,787 (99.78)				410 (99.27)	2,197 (99.68)

REFERENCES

d'Almada, Philip J. and Cynthia Berg (1993). "Linking Databases with Base SAS Software to Facilitate Client/End-user Access to an Extended Data Resource," *Proceedings of the Eighteenth Annual SAS Users Group International Conference*, 18, 228-231.

SAS Institute Inc., *SAS User's Guide: Basics, Version 5 Edition*. Cary, NC: SAS Institute Inc., 1985. 1290 pp.

ACKNOWLEDGEMENTS

The author appreciates the Developmental Disabilities Branch of the Centers for Disease Control and Prevention (CDC), U.S. Department of Health and Human Services (HHS), Atlanta, Georgia, for the resources to prepare this document. The author also appreciates the devotion and support of his wife, Shelley, and his children, Leslie, Seth, Sammy, and Abby.

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries. IBM is a registered trademark or trademark of International Business Machines Corporation. * indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

The author may be reached at the following
postal address:

c/o CDC,
4770 Buford Hwy., F-15,
Atlanta, Georgia 30341-3724

office telephone: 770-488-7283

internet address: pxd2@cehbddd.em.cdc.gov