

Designing the Optimal File Linkage Algorithm (OFLA)

Tsai Mei Lin, Arkansas Department of Health, Little Rock, AR

ABSTRACT

In Maternal and Child Health, file linkage is an increasingly important tool for public health research, particularly when it involves linking birth data to infant death data. Successfully linking these files can lead to improvements in the effectiveness and efficiency of the delivery of public health services. Since linked files are more valuable than the sum of their parts, much more information is available than their constituent databases. Unfortunately, with no common clean and unique identifier, file linkage is much more of an art than a science and tends to involve highly subjective decisions on how to proceed. The objective of this study is to develop multi-step algorithms that use various combinations of variables in order to correctly match as many records as possible, while minimizing false matches. To test the strength of each algorithm, linkage results are compared to a “gold standard” file. Previously constructed, the “gold standard” file is the result of a labor-intensive effort to manually link the infant death file to the birth file. Having a “gold standard” enables the analyst to independently confirm the strength of the algorithms. By identifying the most efficient steps and optimizing their order, and comparing results to the “gold standard”, very efficient file linkage algorithms can be developed using the SAS® system. This paper describes how SAS is used to develop algorithms that minimize false-positives links. Topics covered in this paper include metadata, macro language, and match-merge.

INTRODUCTION

Linkage of large files has traditionally been as much a subjective, qualitative exercise as an objective, statistical endeavor. Typically, an analyst would begin with a few variables felt to have a high degree of “uniqueness” that were relatively “clean.” The analyst would link on three or four of them and then examine the results, with a particular eye to eliminating false positives. Upon completion of the examination, all the true positives are transferred to a separate file. Then the analyst drops one variable and adds another to the algorithm, and repeats the procedure. Given the iterative nature of the process, it is intensive and may involve several dozen steps.

A problem arises with the subjectivity with which decisions are made regarding the utility of the variables and the order in which they are entered into the algorithm. While this ultimately results in a relatively complete linkage, reliability is compromised, as it is difficult to apply the algorithm to other years, databases, or geographic settings. Indeed, experience has shown that a given step may increase the number of true positives and reduce the number of false positives in one year, and that the same step in the identical algorithm may increase the number of false positives even more than the number of true positives it adds. Therefore, what is needed is an empirical technique for developing the algorithm that does not result in false positives.

CONFUSION MATRIX

To this end, there are several statistical measures of how effective a linkage process is. These include but are not necessarily limited to: sensitivity, specificity, positive predictive value, and negative predictive value. Unfortunately, application of these measures assumes that the analyst has full and complete knowledge of all four possible outcomes, shown in Table 1: true positive, true negative, false positive, and false negative. Usually, such information is not available unless the file linkage has already been performed, exhaustively and thoroughly. Such a “gold standard” file has been created, and the four possible outcomes statistics within the confusion matrix in Table 1 are able to be calculated.

Table 1: Confusion Matrix of Possible Outcomes from OFLA

OFLA Result	Gold Standard Result	
	Match	No Match
Match	True Positive <i>Correctly links records</i>	False Positive <i>Does link wrong records</i>
No Match	False Negative <i>Does not link correct records</i>	True Negative <i>Does not link wrong records</i>

DATA SOURCES

To help build the OFLA, three files were used: birth certificate data, infant death data, and a “gold standard” file. These files will help develop the algorithm and test its strength. The birth and infant death files are prepared to be matched to each other programmatically, whose outcomes linkages are then compared against the “gold standard” file. For the purposes of this exercise, the Arkansas Center for Health Statistics uses 17 years of births and infant deaths as the database for the development of the algorithm and a “gold standard” linked database with which to experiment.

BIRTH CERTIFICATE AND INFANT DEATH FILES

The birth certificate file contains 650,705 records of birth events happening between January 1, 1985 and December 31, 2001. The infant death file, which contains records of live-born individuals who die before reaching one year of age, contains 6,193 records from the same time period. Table 2 lists fourteen identifiers, found in both files, which are used, most as combinations, in developing the OFLA.

Table 2: Variables Used in OFLA

Variable	Description
first_name	Baby's First Name
first_sound	Soundex of Baby's First Name
last_name	Baby's Last Name
last_sound	Soundex of Baby's Last Name
mfirst_name	Mother's First Name
mlast_name	Mother's Last Name
mlast_sound	Soundex of Mother's Last Name
maiden_name	Mother's Maiden Name
ffirst_name	Father's First Name
flast_name	Father's Last Name
cityname	City Name
bth_date	Date of Birth
idbrhosp	Hospital
ssn	Social Security Number (SSN)

GOLD STANDARD DATA

A “gold-standard” file was created by manually linking together 650,705 births and 6,193 infant deaths from the Arkansas birth/infant death data system for a 17 year period (1985–2001). The linked file contains two variables: birth certificate number and infant death certificate number. Since multi-year infant deaths was linked to multi-year birth certificates, with careful manual review, the file created was large enough to provide a robust platform for development of the OFLA and an objective basis for measuring its validity.

METHODOLOGY

It takes three steps to design the OFLA: preparing birth/infant death data, deriving metadata, and developing the OFLA. The goal is to develop an OFLA that determines the order that variable

combinations should be used as match keys when running against data not having an associated “gold standard” file. The SAS code in Appendix A, B, and C is used to test, loop, calculate, decide, and order the variable combinations based on the fewest false positives.

STEP 1: PREPARE BIRTH AND INFANT DEATH DATA SETS

The birth/infant death data sets were both scrubbed in order to make them compatible with each other for matching purposes. The major actions involved in data preparation are:

- Clean and subset Files
- Parse and convert personal and city names to Soundex version
- Make variable attributes (i.e. name, type, length and format) consistent across files
- Remove duplicates
- Create unique ID for both files

Analysts need to be very familiar with the databases and the use of the Soundex function. Infant birth date must be clean and accurate in order to produce meaningful results.

STEP 2: DERIVE METADATA

The metadata consists of a list of variables, either alone, or in combination, that will act as match keys when linking together birth and infant death data sets. The variable combinations are chosen from Table 2. It would be ideal to use all single variable, two-variable combinations, three-variable combinations, etc., but this would generate too many possible combinations. It was decided that the twenty-four variable combinations in Table 3 would be used as metadata. Note that Table 3 contains SSN. Later, a second metadata will not contain SSN so that comparisons can be made of an OFLA developed with SSN to one developed without SSN.

Table 3: Variable Combinations used as Metadata in OFLA Development

Dset	Variable Combination
v1a	ssn
v3a	bth_date last_name first_name
v3b	bth_date mlast_name first_name
v3c	bth_date maiden_name first_name
v3d	bth_date last_sound first_name
v3e	bth_date last_sound first_sound
v3f	bth_date mlast_sound first_name
v3g	bth_date mlast_sound first_sound
v4b	bth_date mlast_name first_name cityname
v4d	bth_date last_name cityname idbrhosp
v4e	bth_date last_name first_name idbrhosp
v4f	bth_date last_name first_sound cityname
v5a	bth_date last_name first_name maiden_name mfirst_name
v5b	bth_date last_name first_name mlast_name mfirst_name
v5c	bth_date last_name first_name flast_name ffirst_name
v5d	bth_date last_name first_name cityname idbrhosp
v5e	bth_date maiden_name mlast_name last_name flast_name
v5f	bth_date last_name ffirst_name mfirst_name idbrhosp
v6a	bth_date maiden_name mlast_name last_name flast_name first_name
v7a	bth_date last_name first_name flast_name ffirst_name mlast_name mfirst_name
v7b	bth_date last_name first_name flast_name ffirst_name maiden_name mfirst_name
v7c	bth_date maiden_name mlast_name last_name flast_name first_name mfirst_name
v8a	bth_date last_name first_name flast_name ffirst_name mlast_name mfirst_name idbrhosp
v8c	bth_date maiden_name mlast_name last_name flast_name first_name mfirst_name ffirst_name

In Table 3, the Dset values identify a particular variable combination, with the second byte indicating the number of variables. For example, the “v7” in “v7a” represents the class of seven-variable combinations.

STEP 3: DEVELOP OPTIMAL FILE LINKAGE ALGORITHM

Using comparable variables from the birth and infant death certificates, a SAS program with a “looping” macro tested the 24 separate variable combinations in Table 3. The macro then removed the step that generated the fewest false positives from the program and wrote out the linked records to another file. Repeating the process, the macro tested the remaining 23 steps against the remaining data and, once again, removed the optimal step and linked data. This continued until all steps were exhausted. The method is described two ways: anecdotally and programmatically.

Anecdotal Description of OFLA Development

Each data set is sorted and match-merged on the first variable combination in the metadata in Table 3 (SSN). The matches are then sorted on certificate number and match-merged against the “gold standard” file. Confusion matrix statistics are calculated and stored. Next, each dataset is sorted and match-merged on the second variable combination in the metadata in Table 3 (bth_date last_name first_name). The matches are then sorted on certificate number and match-merged against the “gold standard” file. Confusion matrix statistics are calculated and appended to the first set of confusion matrix statistics. This process repeats on each of the remaining 22 variable combinations in Table 3. After passing through all variable combinations in Table 3 the first time, the variable combination yielding the fewest false positives is removed from the metadata, leaving 23 variable combinations in which to use for the next pass, and its linked data is written to a master file. After repeating the just-described procedure on the 23 remaining variable combinations, the variable combination yielding the fewest false positives is removed from the metadata and its linked data is appended to the master file, thereby leaving 22 variable combinations with which to repeat the procedure. Again, after the next pass, the variable combination yielding the fewest false positives is removed from the metadata, leaving 21 variable combinations with which to repeat the procedure, and its linked data are appended to the master file. The procedure repeats until all 24 passes are complete. Overall, the variable combination yielding the fewest false positives after the first pass is considered the best variable combination. The variable combination used in the last pass is considered the worst variable combination.

Programmatic Description of OFLA Development

The OFLA method can be described while citing the programs found in Appendix A, B, and C. The number of variable combinations in the original metadata is represented by $r = 24$. This also means that there will be r passes through the metadata. In each pass, there is a test consisting of a match-merge of the birth and infant death data sets, whose matches are then match-merged against the “gold standard” file. When the pass ends, the best variable combination is removed from the metadata, its linked data written to a master file, and then the next pass begins. If j represents a particular pass and i a particular test within a pass, then $i = 1$ to $r - j + 1$, where $j = 1$ to r . In the program, the indexes j and i have macro variables &numcomb and &nobs, respectively, as upper bounds, shown in Table 4.

Table 4: Macro Variables used as Upper Bounds for Indexing

Macro Variable	Value	Location Within Code
&numcomb	r	Appendix A, Line 8
&nobs	$r - j + 1$	Appendix A, Line 49; Appendix C, Lines 13, 75

Appendix A contains the program that calls on line 53 Appendix B, which is the macro that match-merges the birth and infant death data sets and then match-merges those matches against the “gold standard” file. On line 60, the code in Appendix C is called, which will perform all r passes and select the best variable combination after each pass.

RESULTS

The OFLA development was performed separately twice: one development including SSN and the other excluding SSN.

WITH SOCIAL SECURITY NUMBER

After applying the SAS code to the metadata in Table 3, which includes SSN, Table 5 displays the variable combinations ordered from best to worst based on the minimum number of false positives calculated after each pass.

Table 5: OFLA Counts with Social Security Number

Dset	Variable Combination	Pass Number	Process Time	Match Count	False Positive Count
v1a	ssn	1	1:24:33	536	0
v8c	bth_date maiden_name mlast_name last_name flast_name first_name mfirst_name ffirst_name	2	1:19:02	138	4
v7b	bth_date last_name first_name flast_name ffirst_name maiden_name mfirst_name	3	1:14:06	16	1
v7c	bth_date maiden_name mlast_name last_name flast_name first_name mfirst_name	4	1:11:19	9	2
v6a	bth_date maiden_name mlast_name last_name flast_name first_name	5	1:06:45	16	1
v8a	bth_date last_name first_name flast_name ffirst_name mlast_name mfirst_name idbrhosp	6	1:02:37	372	2
v5e	bth_date maiden_name mlast_name last_name flast_name	7	0:59:11	24	4
v7a	bth_date last_name first_name flast_name ffirst_name mlast_name mfirst_name	8	0:56:04	725	5
v5a	bth_date last_name first_name maiden_name mfirst_name	9	0:51:23	774	6
v3c	bth_date maiden_name first_name	10	0:47:33	204	1
v5b	bth_date last_name first_name mlast_name mfirst_name	11	0:47:11	460	1
v4b	bth_date mlast_name first_name cityname	12	0:43:11	233	2
v3b	bth_date mlast_name first_name	13	0:38:14	41	2
v3f	bth_date mlast_sound first_name	14	0:36:54	48	1
v3g	bth_date mlast_sound first_sound	15	0:32:30	201	6
v5d	bth_date last_name first_name cityname idbrhosp	16	0:29:57	834	6
v4e	bth_date last_name first_name idbrhosp	17	0:27:48	82	1
v5f	bth_date last_name ffirst_name mfirst_name idbrhosp	18	0:22:45	51	2
v5c	bth_date last_name first_name flast_name ffirst_name	19	0:18:49	962	5
v3a	bth_date last_name first_name	20	0:17:04	261	2
v3d	bth_date last_sound first_name	21	0:13:51	19	0
v4f	bth_date last_name first_sound cityname	22	0:11:20	58	0
v3e	bth_date last_sound first_sound	23	0:07:35	21	1
v4d	bth_date last_name cityname idbrhosp	24	0:03:05	22	5
TOTAL			16:42:47	6107	60

Overall, the true positive rate was 97.46% and the false positive rate was 0.807%, while linking 6107 infant death records. In the first pass, which tested all twenty-four variable combinations, SSN was the variable selected that yielded the fewest false positives. Generally, original names performed better than their Soundex versions and, except for SSN, combinations having many variables performed better than combinations having few variables. Linkages involving mother's, father's and infant's name together performed better than subsets of these names.

WITHOUT SOCIAL SECURITY NUMBER

Table 6 shows the results of the OFLA developed when SSN was excluded from the metadata, thereby using twenty-three variable combinations.

Table 6: OFLA Counts without Social Security Number

Dset	Variable Combination	Pass Number	Process Time	Match Count	False Positive Count
v8c	bth_date maiden_name mlast_name last_name flast_name first_name mfirst_name ffirst_name	1	1:20:28	139	4
v7b	bth_date last_name first_name flast_name ffirst_name maiden_name mfirst_name	2	1:15:16	16	1
v7c	bth_date maiden_name mlast_name last_name flast_name first_name mfirst_name	3	1:12:24	9	2
v6a	bth_date maiden_name mlast_name last_name flast_name first_name	4	1:07:18	16	1

Dset	Variable Combination	Pass Number	Process Time	Match Count	False Positive Count
v8a	bth_date last_name first_name flast_name ffirst_name mlast_name mfirst_name idbrhosp	5	1:04:10	389	2
v5e	bth_date maiden_name mlast_name last_name flast_name	6	1:00:53	25	4
v7a	bth_date last_name first_name flast_name ffirst_name mlast_name mfirst_name	7	0:56:52	777	5
v5a	bth_date last_name first_name maiden_name mfirst_name	8	0:53:40	840	6
v5b	bth_date last_name first_name mlast_name mfirst_name	9	0:48:33	524	1
v3c	bth_date maiden_name first_name	10	0:47:57	220	2
v4b	bth_date mlast_name first_name cityname	11	0:42:48	254	2
v3b	bth_date mlast_name first_name	12	0:39:10	43	2
v3f	bth_date mlast_sound first_name	13	0:38:23	53	1
v3g	bth_date mlast_sound first_sound	14	0:34:11	214	6
v5d	bth_date last_name first_name cityname idbrhosp	15	0:30:07	869	6
v4e	bth_date last_name first_name idbrhosp	16	0:28:04	87	1
v5f	bth_date last_name ffirst_name mfirst_name idbrhosp	17	0:24:46	52	2
v5c	bth_date last_name first_name flast_name ffirst_name	18	0:24:27	1141	5
v3a	bth_date last_name first_name	19	0:18:03	294	2
v3d	bth_date last_sound first_name	20	0:15:28	22	0
v4f	bth_date last_name first_sound cityname	21	0:13:37	67	0
v3e	bth_date last_sound first_sound	22	0:08:46	23	1
v4d	bth_date last_name cityname idbrhosp	23	0:03:50	22	5
TOTAL			15:49:11	6096	61

Overall, the true positive rate was 97.27% and the false positive rate was 0.824%, while matching to 6096 infant death records. Similar results were achieved when comparing the SSN-inclusive OFLA to the SSN-exclusive OFLA. Records originally linked on SSN could be linked using other variable combinations when SSN was not present. Only eleven fewer matches were attained when SSN was excluded from the process. Excluding SSN caused the variable combination v5c to “pick up the most slack” by adding 179 matches. The linkage variable combinations having the fewest false positives were selected in about the same order as they were when developing the algorithm with SSN. Only those variable combinations indicated by v3c and v5b switched order. Not having SSN in the algorithm had virtually no impact on the overall linkage rates, only decreasing the true positive rate 0.19% and increasing the false positive rate 0.017%.

SUMMARY AND RECOMMENDATIONS

File linkage is an increasingly important tool for public health research, which can lead to improvements in the effectiveness and efficiency of the delivery of public health services. By identifying the most efficient steps and optimizing their order, very efficient file linkage algorithms can be developed using the SAS program found in the appendices.

A “gold standard” file was used to help calculate all confusion matrix statistics, in particular, the number of false positives. However, the other elements could be used in a further capacity. For example, after a pass, if two variable combinations are tied for the fewest false positives, instead of choosing the first one alphabetically, like in this study, the SAS program could be altered to choose the variable combination yielding the highest number of true positives.

True positive rates were very high, at least 97%, using either SSN-inclusive or SSN-exclusive metadata. However, future study could research how all confusion matrix statistics are impacted when using either all possible variable combinations, if feasible, or a much larger metadata data set.

Variation in the cleanliness or completeness of the birth/infant death data could significantly impact data preparation methods, metadata derivation and ultimately OFLA development. Also, data inconsistency or interventions for an OFLA developed in one time period could significantly affect outcomes when applied to data from a different time period. Therefore, it is probably necessary to develop several OFLAs in order to adapt to changing data conditions, which means manually reviewing at least some of the linked records, especially from those steps that generate relatively high false positive rates.

REFERENCES

Douglas R. Murray (2003), "Workplan for Birth-Death File Linkage", Arkansas Department of Health, Center for Health Statistics, Little Rock, Arkansas

SAS Institute, Inc. (1990), *SAS® Guide to Macro Processing, Version 6, Second Edition* Cary, NC. SAS Institute Inc.

SAS Institute, Inc. (1990), *SAS® Language: Reference, Version 6, First Edition* Cary, NC. SAS Institute Inc.

SAS Online Doc Version 8

ACKNOWLEDGEMENTS

Special thanks to ...

Tom Hotard Acxiom Corporation

Douglas R. Murray Former Director of Center for Health Statistics, Arkansas Department of Health

for their invaluable guidance and suggestions.

TRADEMARK CITATIONS

SAS and other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

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

CONTACT INFORMATION

Tsai Mei Lin
Arkansas Department of Health
Center for Health Statistics
4815 W. Markham St., Slot 19
Little Rock, AR 72205-3867
tlin@healthyarkansas.com

Appendix A

Code used to create the optimal file linkage for births and infant death data

```
1  options nocenter ps=80 ls=200;
2  libname library 'C:\Projects\Request\VRD-Research';
3  libname stat ' C:\Projects\Request\VRD-Research\Gordon\TML-Test\NewData';
4
5  %let dsetbth=stat.bths_ssn_e;
6  %let dsetdth=stat.inf8501ssn_e;
7  %let goldstd=stat.main;
8  %let numcomb=24;
9
10 data metadata;
11     infile cards ls=200 truncover;
12     input @1 dset $char3. @6 varlist $char120.;
13 cards;
14 v1a  ssn
15 v3a  bth_date last_name first_name
16 v3b  bth_date mlast_name first_name
17 v3c  bth_date maiden_name first_name
18 v3d  bth_date last_sound first_name
19 v3e  bth_date last_sound first_sound
20 v3f  bth_date mlast_sound first_name
21 v3g  bth_date mlast_sound first_sound
22 v4b  bth_date mlast_name first_name cityname
23 v4d  bth_date last_name cityname idbrhosp
24 v4e  bth_date last_name first_name idbrhosp
25 v4f  bth_date last_name first_sound cityname
26 v5a  bth_date last_name first_name maiden_name mfirst_name
27 v5b  bth_date last_name first_name mlast_name mfirst_name
28 v5c  bth_date last_name first_name flast_name ffirst_name
29 v5d  bth_date last_name first_name cityname idbrhosp
30 v5e  bth_date maiden_name mlast_name last_name flast_name
31 v5f  bth_date last_name ffirst_name mfirst_name idbrhosp
32 v6a  bth_date maiden_name mlast_name last_name flast_name first_name
33 v7a  bth_date last_name first_name flast_name ffirst_name mlast_name mfirst_name
34 v7b  bth_date last_name first_name flast_name ffirst_name maiden_name mfirst_name
35 v7c  bth_date maiden_name mlast_name last_name flast_name first_name mfirst_name
36 v8a  bth_date last_name first_name flast_name ffirst_name mlast_name mfirst_name idbrhosp
37 v8c  bth_date maiden_name mlast_name last_name flast_name first_name mfirst_name ffirst_name
38 ;
39 run;
40
41 data firstmetadata;
42     set metadata;
43 run;
44
```

```

45 data assign;
46     set metadata end=lastobs;
47     call symput('dset'||compress(_n_),compress(dset));
48     call symput('vlist'||compress(_n_),left(trim(varlist)));
49     if lastobs then call symput('nobs',compress(_n_));
50 run;
51
52 /* Macro Link found in Appendix B to link births against infant deaths data */
53 %include ' C:\Projects\Request\VRD-Research\SendOut-Final\link_SSN.mac ';
54
55 proc datasets nolist;
56     delete allmatches runmatches timeinfo;
57 run;
58
59 /* Macro Loop found in Appendix C to create iteration run based on metadata */
60 %include ' C:\Projects\Request\VRD-Research\SendOut-Final\loop_SSN_E.mac ';
61
62 proc freq data = allmatches noprint;
63     tables dset/out=statmatch(drop=percent rename=(count=match_count));
64 run;
65
66 data timing;
67     set timeinfo;
68     begin_date=put(ddate1,datetime.);
69     end_date=put(ddate2,datetime.);
70     diff=put(total,time.);
71 run;
72
73 proc sort data=firstmetadata out=first;
74     by dset;
75 run;
76 proc sort data=timing out=time;
77     by dset;
78 run;
79 proc sort data=statmatch out=stat;
80     by dset;
81 run;
82
83 data subset(keep=dset count rename=(count=fp_count));
84     set runmatches;
85 run;
86 proc sort data=subset out=fp;
87     by dset;
88 run;
89
90 data report(drop=total rename=(diff=Process_Time));
91     merge first time stat fp;
92     by dset;

```

```
93      run;
94      proc sort data=report;
95          by runnum;
96      run;
97
98      proc contents data = report;
99      run;
100     proc print data = report;
101         var dset varlist runnum Process_Time match_count fp_count;
102         sum match_count fp_count;
103     run;
104
105     proc means data = timeinfo;
106         var total;
107         output out=out1 sum(total)=sum_total;
108         format total time.;
109     run;
110     proc print data = out1;
111     run;
```

Appendix B

Code used to link births against infant death data based on metadata

```

1  %macro link(dsetbth,dsetdth,macvar,doutmat,doutbth,doutdth);
2  proc sort data=&dsetbth;
3      by &macvar;
4  run;
5  proc sort data=&dsetdth;
6      by &macvar;
7  run;
8
9  data &doutmat(keep=b_certno id_bcertno id_dcertno dset)
10      &doutbth(drop=bstate dcfirst_ dcity dclast_ dcmidd_ dcnty dctype_ dffirst_
11          dflast_ dfmidd_ dftitle_ dhosp_ dmfirst_ dmlast_ dmmidd_ dmtitle_
12          drcity drcnty drstate dsex dstate hospid id_bcertno id_bdate id_dcertno
13          id_ddate id_dname id_fname id_mname infd_id yearb yeard socsec)
14      &doutdth(drop=dob_ fssn_ hcode_ maiden mssn_ ocity ocounty ostate rcity rcounty
15          rstate b_bdate b_certno b_cname b_fname b_mname bcfirst_ bclast_ bcmidd_
16          bctitle_ bffirst_ bflast_ bfmidd_ bftitle_ bmfirst_ bmlast_ bmmidd_
17          bmtitle_ brth_id year ssn);
18  merge &dsetbth(in=a) &dsetdth(in=b);
19  by &macvar;
20  length dset $3;
21  dset="&&dset&i";
22  if a and b then output &doutmat;
23  else if a and not b then output &doutbth;
24  else if b and not a then output &doutdth;
25  run;
26
27  proc sort data=&goldstd out=gstand;
28      by dcertno certno;
29  run;
30  proc sort data=&doutmat out=s_match(rename=(id_dcertno=dcertno b_certno=certno));
31      by id_dcertno b_certno;
32  run;
33
34  data dummy1;
35      merge gstand(in=a) s_match(in=b);
36      by dcertno certno;
37      if a and b then match=1;
38      if a and not b then match=2;
39      if not a and b then match=3;
40      if not a and not b then match=4;
41  run;
42  proc freq data = dummy1 noprint;
43      tables match/out=freqcnt(drop=percent);

```

```
44      run;
45
46      data template;
47          length match count 5;
48          match=1; count=0; output;
49          match=2; count=0; output;
50          match=3; count=0; output;
51          match=4; count=0; output;
52      run;
53
54      data info_&&dset&i;
55          merge template freqcnt;
56          by match;
57          dset="&&dset&i";
58      run;
59      %mend link;
```

Appendix C

Code used to create iteration run based on metadata set

```
1  %macro execute;
2  %do j = 1 %to &numcomp;
3      proc datasets nolist;
4          delete dinfo&j;
5      run;
6
7      data time1;
8          length runnum 5;
9          runnum="&j"+0;
10         ddate1=datetime();
11     run;
12
13     %do i = 1 %to &nobs;
14         %link(dsetbth=&dsetbth, dsetdth=&dsetdth, macvar=&&vlist&i, doutmat=dmata&&dset&i,
15             doutbth=dbth&&dset&i, doutdth=infda&&dset&i);
16
17     proc datasets nolist;
18         append base=dinfo&j data=info_&&dset&i;
19     run;
20     %end;
21
22     data time2;
23         length runnum 5;
24         runnum="&j"+0;
25         ddate2=datetime();
26     run;
27
28     data meta1;
29         merge metadata dinfo&j;
30         by dset;
31     run;
32
33     proc sort data = meta1(where=(match=3)) out=s_meta1;
34         by count;
35     run;
36
37     data newmeta(drop=runnum) bestmatch;
38         set s_meta1;
39         length runnum 5;
40         if _n_=1 then do;
41             call symput('doutmat','dmata'||compress(dset));
42             call symput('dsetbth','dbth'||compress(dset));
43             call symput('dsetdth','infda'||compress(dset));
```

```

44     call symput('dsetpick',compress(dset));
45     will_delete=1; runnum="&j"+0; output bestmatch;
46     end;
47     else will_delete=0;
48     output newmeta;
49 run;
50
51 data combine_time;
52     merge time1 time2;
53     by runnum;
54     total = ddate2-ddate1;
55     dset="&dsetpick";
56 run;
57
58 proc datasets nolist;
59     append base=timeinfo data=combine_time;
60 run;
61 proc datasets nolist;
62     append base=allmatches data=&doutmat;
63 run;
64 proc datasets nolist;
65     append base=runmatches data=bestmatch;
66 run;
67 proc sort data = newmeta(where=(will_delete=0))
68     out=metadata; by dset;
69 run;
70
71 data assign;
72     set metadata end=lastobs;
73     call symput('dset'||compress(_n_),compress(dset));
74     call symput('vlist'||compress(_n_),left(trim(varlist)));
75     if lastobs then call symput('nobs',compress(_n_));
76 run;
77 %end;
78 %mend execute;
79 %execute;

```