

One method for improving quality of data in longitudinal study

Milorad Stojanovic
Ruth Heuer
RTI International
Education Survey Studies Division
RTP, North Carolina

ABSTRACT:

Data quality is of utmost importance in every project. The quality of identification variables (key variables) is especially important in longitudinal studies in order to match individuals correctly across years which, in turn, allows for analysis over time. In improving quality of data, all available cost effective sources were used: data from previous years, financial aid data, collections of last and first names. Cleaning process used several stages to improve cases with no SSN and imprecise last name, first name, and date of birth. Macro code was developed for repetitive tasks during matching process.

KEYWORDS

Longitudinal study, data matching, data quality.

INTRODUCTION

In the US Department of Education's Upward Bound study, students are tracked over several years of participation in an Upward Bound project. Data on each participant and prior year participant is submitted by the project on an annual basis, and a longitudinal file was created to allow analysis of outcomes such as high school graduation and college enrollment. In the initial attempt to create a longitudinal file, the variables used for matching participants from one year to the next were PR (project code) and SSN (Social Security Number) of participants.

The PR variable was reported properly in the data. However, the second key variable, SSN, was not always reported properly because SSN was reported by HS (High School) students who frequently do not know their SSN. While projects provided the correct value or at least consistent value of SSN for most participants from year to year, it was not unusual for the project to report different SSNs for the same student in different years. These differences ranged from one digit incorrectly keyed to completely different values. Common SSN mistakes included transposed adjacent digits, shifted content of SSN one position to the left or to the right and the same SSN reported for two different participants. Also a small percentage provided obviously 'bogus' values for their SSNs or entered all zeros or all 9s or simply left that variable blank.

Because of these problems with the SSN variable, the matching resulted in a longitudinal file with more 'unique' participants than there were in reality as well as a small number of false matches. To correct for this, we devised a series of steps aimed not only at cleaning SSN, but also using first name, last name and date of birth to properly match a participant's data across years. Each of those additional identifying variables brought its own problem and required a unique solution.

Cleaning variables first name (FIRSTNM) and last name (LASTNM).

There were two problems.

1. There were different standard in filling out name variables between years (All capital letters, mix of capital letters and small letters).
2. In the number of cases name variables had special characters (mainly Spanish, Portuguese, but also real special characters).

First problem was easily resolved – we converted all name variables to capital letters. Second problem was at least twofold problem. Identification of special Spanish and Portuguese characters was relatively easy; we were able to consult with colleagues fluent in these languages. Identification of special characters like # or * etc. in name variables required more analysis; we built two 'data bases' for last names and for first names to get better feeling which is the best replacement for special characters. Also we removed extraneous Jr, III, etc mention removing the extraneous Jr, III, etc that may be entered slightly differently from year to year.

Cleaning Date of Birth variable (DOB)

If DOB was unknown during the data entry it was set to 01/01/9999. Our target population was high school students so we set limits of 01/01/1950 to 12/31/1990 (based on examination of the data which showed that DOB values for year were typically in the range of 197x to 198x). If the date of birth was out of that range we assigned missing value for that case.

Also we found a number of cases where date of birth was 01/01/1960. Upon further investigation we discovered that projects were instructed to assign a value of 0 to date of birth when DOB was unknown.

Cleaning SSN

If SSN was obviously invalid (i.e., consisted of too many digits, too few digits, all digits the same, negative numbers, special codes like '-3', '-4', as well as SSN starting with '9') we set the SSN to blank or missing. The original SSN was saved in SSN_OLD for future matching (in the event that the same invalid SSN was used in another year).

Number of records with missing SSN per year

Year	Total # of records	# of valid records	# of records with blank SSN	%
2000-01	77,422	76,722	700	0.90
2001-02	91,353	90,370	983	1.08
2002-03	106,749	105,624	1,125	1.05
2003-04	128,837	127,175	1,662	1.29
Total	404,361	439,891	4,470	1.11

Replace missing SSN with valid SSN.

To replace the missing SSNs with valid SSN we used two sources:

1. Financial aid database.
2. Other years of Upward Bound data.

If a participant's SSN was blank (either because it was not provided or it was deemed invalid), we attempted to find the participant on a financial aid file using last name, first name and date of birth. The financial aid file included SSN, last name, first name, and DOB for students who had filed a Free Application for Federal Student Aid (FAFSA) or received a Pell grant. Matching with financial DB was on PR, LASTNM, FIRSTNM, and DOB. If a match was found, the SSN from the financial aid file was assigned to replace the blank SSN on our file.

Here are the steps taken to match against financial aid file:

1. For each year of data, we separated the data set into two files – one with acceptable SSN and other with current blank SSN ('NO_SSN_YEARn').
2. After that we tried to the observation in 'NO_SSN_YEARn' data sets with Financial data set by PR, LASTNM, FIRSTNM and DOB.

* Matching cases with missing SSN against Federal Financial Aid data ;

* Year 2003 ;

```
data ub_LONG_2003 ub_LONG_2003_NoSSN ;
  set ub_LONG_2003 ;
  if SSN = '' then output ub_LONG_2003_NoSSN ;
  else          output ub_LONG_2003 ;
run ;
```

```

proc sort data = ub_LONG_2003_NoSSN ;
  by pr LASTNM FIRSTNM DOB ;
run ;

proc sort data = rl.ubfin(keep=pr ssn LASTNM FIRSTNM DOB rename=(ssn=ssn1))
  out = ubfin ;
  by pr LASTNM FIRSTNM DOB ;
run ;

data ubfin ubfindupl ;
  set ubfin ;
  by pr LASTNM FIRSTNM DOB ;
  if not (first.dob and last.dob) then output ubfindupl ;
  if last.dob then output ubfin ;
run ;

proc print data=ubfindupl ;
title "List of duplicates from UBFIN on PR, LASTNM, FIRSTNM, and DOB" ;
var pr SSN1 LASTNM FIRSTNM DOB ;
run ;

data ub_LONG_2003_match ub_LONG_2003_NoMatch(drop=ssn1) ;
  merge ub_LONG_2003_NoSSN(in=in1) ubfin(in=in2) ;
  by pr LASTNM FIRSTNM DOB ;
  if in1 and in2 then do ;
    ssn = ssn1 ;
    output ub_LONG_2003_match ;
  end ;
  else if in1 and not in2 then output ub_LONG_2003_NoMatch ;
run ;

```

Number of cases with 'bad' SSN which were corrected using FAFSA data

Year	Total # of records	# of corrections	%
2000-01	935	126	13.48
2001-02	1,316	109	8.28
2002-03	1,322	37	2.80
2003-04	1,837	97	5.28

If, after matching against the financial aid database, the SSN was still invalid, we attempted to find the same participant on another year's Upward Bound data. We had 4 years of participant data, so each case with missing SSN in a given year was compared to each of the other 3 years of participant data using PR, LASTNM, FIRSTNM, and DOB. If an exact match was found, the SSN from the other year's file was assigned to replace the blank SSN.

* Checking if full matching exist ;

```

proc sort data=Ub_LONG_2002(keep=pr ssn LASTNM FIRSTNM DOB)
  out = ub2002_sort(compress=yes) ;
  by pr SSN LASTNM FIRSTNM DOB ;
run ;

data match2002 nomatch2002 ;
  merge nomatch2003(in=in1) ub2002_sort(in=in2) ;
  by pr SSN LASTNM FIRSTNM DOB ;
  if in1 and in2 then output match2002 ;
  else if in1 and not in2 then output nomatch2002 ;
run ;

```

```
proc print data = match2002 ;
title "Listing of UBFIN duplicate cases which matched with UB/UBMS files - Year2" ;
var pr SSN LASTNM FIRSTNM DOB ;
run ;
```

* Year 2001 ;

```
proc sort data=Ub_LONG_2001(keep=pr ssn LASTNM FIRSTNM DOB)
out = ub2001_sort(compress=yes) ;
by pr SSN LASTNM FIRSTNM DOB ;
run ;
```

```
data match2001 nomatch2001 ;
merge nomatch2002(in=in1) ub2001_sort(in=in2) ;
by pr SSN LASTNM FIRSTNM DOB ;
if in1 and in2 then output match2001 ;
else if in1 and not in2 then output nomatch2001 ;
run ;
```

```
proc print data = match2001 ;
title "Listing of UBFIN duplicate cases which matched with UB/UBMS files - Year1" ;
var pr SSN LASTNM FIRSTNM DOB ;
run ;
```

[List the number and percent of cases with missing SSN prior to this step and the number for which we got an SSN from other year's Upward Bound data].

If SSN was still blank after matching against the financial aid file and Upward Bound data from other years, a dummy SSN in the range of 909090001 to 909097999 was assigned for identification purposes.

For cases where SSN was a duplicate (within a project) but LASTNM, FIRSTNM, and DOB are different (i.e., the same SSN was reported for two different participants) then assign a dummy SSN in the range of 9090980001 to 909099999 to both records and retain the reported SSN in SSN_OLD.

Matching across years to create the longitudinal dataset

1. First, using the strictest matching criteria, we matched on PR, SSN, LASTNM, FIRSTNM, DOB.
2. For those that did not match, we loosened the matching criteria, but not so much that we let in false matches
 - PR, LASTNM, FIRSTNM, DOB
 - PR, LASTNM, FIRSTNM, plus either "similar birth date" (defined as month and year match and day is within 10 days OR day and year match and month is within 1 OR month and day match and year is within 1 OR year matches and month/day are transposed) or partial SSN (7 out of 9 digits match)
 - PR and "similar birth date" and partial SSN plus either LASTNM or FIRSTNM
 - PR, plus transposed LASTNM and FIRSTNM

* Resolving problem of duplicates within the same year ;

```
proc sort data = UB2002(compress=YES) ;
by pr ssn lastNM firstNM dob ;
run ;
```

```
data UB2002(compress=YES) rl.UB2002DUPLpreY2(compress=YES);
set UB2002 ;
by pr ssn lastNM firstNM dob ;
if not (first.dob and last.dob) then output rl.UB2002DUPLpreY2 ;
else output UB2002 ;
run ;
```

```
data UB2002DUPLlast(compress=YES) rl.UB2002dupl_afterY2(compress=YES);
set rl.UB2002DUPLpreY2 ;
by pr ssn lastNM firstNM dob ;
if not last.firstNM then output rl.UB2002dupl_afterY2 ;
if last.firstNM then output UB2002DUPLlast ;
run ;
```

* Adding deduplicate observations to main file ;

```

data UB2002(compress=YES);
set UB2002 UB2002DUPLlast ;
by pr ssn lastNM firstNM dob ;
run ;

```

*Number of duplicate and de-duplicate records
on PR SSN LASTNM FIRSTNM DOB*

Year	# of duplicate s	# of dropped records	# of dedup records	% of dedup record s
2000-01	1,429	781	648	45.35
2001-02	4	2	2	50.00
2002-03	3	2	1	33.33
2003-04	184	92	92	50.00
Total	1,620	877	743	45.86

Number of duplicate records removed from each year

Year	# of removed records	Total # of records	%
2000-01	17	77422	0.02
2001-02	9	91353	0.01
2002-03	15	107E3	0.01
2003-04	18	129E3	0.01
Total	59	404E3	0.02

*Number of Upward Bound projects, participants and
participants per projects*

Year	Min # of students per project	Max # of students per project	# of projects	# of students
2000-01	18	329	826	77,422
2001-02	22	343	839	91,353
2002-03	45	417	840	106,749
2003-04	16	459	899	128,837

CONCLUSION

Through this process of continuous cleaning of data, first year by year, and later by comparison data from one year by other years, we achieved better quality of data and more correct merges. It gave us valuable experience for building on longitudinal file in the coming years.

DISCLAIMER

Opinions and suggestions on how to improve quality of data stated in the paper are not necessarily the opinions and suggestions of Education Survey Studies Division (RTI) and US Department of Education.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Milorad Stojanovic
RTI International
Education Survey Studies Division
3040 Cornwallis Rd
RTP NC 27709
Work Phone: 919-541-7376
e-mail: milorad@rti.org

Ruth Heuer
RTI International
Education Survey Studies Division
3040 Cornwallis Rd
RTP NC 27709
Work Phone: 919-541-6457
e-mail: heuer@rti.org

SAS and all 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 trademarks of their respective companies.