Record Linkage in Stata

presented at:
North American Stata Users Group Meeting
August 13, 2007
Boston, MA

Presented by:
Michael Blasnik
M. Blasnik & Associates

What is Record Linkage?

- Record linkage involves matching records from two different data files that do not share a unique and reliable key field
  - Jon Smithfield 314 Longwood Drive Boston
  - John Smithfeld 314 Longword Dr Boston
- People are good at assessing matches, but underlying logic is tricky to codify

Record Linkage Methods

- Rules-based
  - enumerate a set of rules for how to decide on matches
  - unwieldy, custom solution needed for each problem
- Probabilistic Record Linkage
  - Fellegi and Sunter "A Theory for Record Linkage" JASA Vol. 64 (1969), 1183-1210.
    - Developed framework based on concepts like P(xvar matches | true record match), P(xvar matches | records not a match) to assess overall probability of a match between records based on matching/mismatching of multiple variables.

Record Linkage Issues

- Assigning Match/Non-match Probabilities
  - By variable and/or observation, based on uniqueness or judgment
    - greater weight to matching on phone # than city
    - greater weight to matching Blasnik than Smith
- Imprecise matches for strings
  - String comparators assess degree of matching
    - Edit distance: # edits needed to transform one string into the other
    - Bigram: proportion of 2 character sub-strings in common
- Speed problem
  - N1*N2 comparisons required for exhaustive search (like nnmatch)
  - Most systems employ "Or-blocking" and/or multiple passes
    - Only select records that match on at least one variable
  - Can also employ large indexed bigram tree for large repeated searches
- Some Matches Unclear
  - 3 groups: matched, unmatched, maybe matched but needs review

Data Cleaning & Prep

- Matching improves if data are properly prepared
  - Addresses
    - Fully standardized addresses best (mass-mailing software)
    - Use standardized abbreviations (Street = St, etc.)
    - Create separate variable for house # since errors less frequent
    - Zip/Postal code standardization (5 digit vs. 9 digit zip code)
  - Dates: separate month, day and year since errors often occur in only one
  - Names
    - Standardize Junior = Jr, etc.
    - Can create phonetic coding version (e.g., soundex)
  - Telephone numbers: separate area code to deal with missing
  - Numeric Data
    - generally better to convert numeric fields to string if you think typographical errors may exist so that string comparators are used

Stand-alone Linkage Systems

- Some free record linkage software
  - Link Plus
    - US CDC free software designed for working with cancer registries, but can be used more widely
  - Febrl
    - open-source application in Python / C by Australian National University

reclink.ado

- Stata ado file to implement basic record linkage
  - User-assigned match and non-match weights per variable
  - Or-Blocking: allowed, automatic if >=4 variables
  - And-Blocking – required exact matches may be specified
  - Bigram string comparator (option to override)
    - user-assignable matching threshold, default =0.6
    - proportional match/non-match weighting
  - Multiple Passes facilitated with –exclude- option
  - Implemented in Stata 8.2 ado code
    - speed benefit from moving to Mata?

help for **reclink**

Record Linkage

reclink *varlist* **using** *filename* , **idmaster**(*varname*) **idusing**(*varname*) **gen**(*newvarname*) [
wmatch(*match weight list*) **wnomatch**(*non-match weight list*) **orblock**(*varlist*)
required(*varlist*) **exclude**(*filename*) **exactstr**(*varlist*) **uvarlist**(*varlist*)
_merge(*newvarname*) **uprefix**(*text*) **minscore**(#) **minbigram**(#) **strict**]

Description

reclink uses record linkage methods to match records between two datasets when no perfect key fields exist. Essentially, it makes a fuzzy merge between two datasets. **reclink** allows for user-defined matching and non-matching weights for each variable in the match and employs a bigram string comparator to assess imperfect string matches.

The user must specify the names of unique identifiers in the master and using datasets to track the matching records. The user must also specify the name of a new variable to hold the matching score (scaled 0-1) for each observation.

To enhance speed, the user may specify variables for or-blocking, which requires at least one field to match perfectly. Or-blocking is the default if 4 or more variables are specified.

reclink: Typical Usage

```
reclink ssn fname lname address phone using fapdata,
wmatch(10 3 6 20 10) wnomatch(7 5 5 15 2)
idmaster(trackid) idusing(fapid) gen(fapscore)
```

- compares datasets based on 5 variables
- since # vars>3, uses or-blocking to only assess using data obs that match perfectly on at least one variable. orblock(none) would over-ride
- by default, bigrams used with a minimum match threshold of 0.6
- match weights are highest for address and ssn
- non-match weight =2 for phone since #s change, but match weight=10
- resulting dataset contains all original obs with a -joinby- to using dataset observations with best scores, if score>0.8 default minimum.
- _merge and fapscore (holding the matching score) are created
- New variables also created: Ussn, Ufname, etc.

Example

mydata.dta

```
fname, lname, address, city, idm
Jon, Smithfield, 314 Longwood Drive, Boston, 101
```

ludata.dta

```
fname, lname, address, city, idlu
John, Smithfeld, 314 Longword Dr, Boston, 1002
```

```
reclink  address fname lname city using ludata ,
  gen(myscore) idmaster(idm) idusing(idlu)
```

- they match! myscore = 0.9964, _merge=3, all fields in both shown
  - can then drop U* after browsing to confirm the match
- help file explains more details about setting options and specifying weights