dtalink

Faster probabilistic record linking and deduplication methods in Stata for large data files

Presentation at the 2018 Stata Conference

Columbus, Ohio

July 20, 2018

Keith Kranker

Abstract

Stata users often need to link records from two or more data files, or find duplicates within data files. Probabilistic linking methods are often used when the file(s) do not have reliable or unique identifiers, causing deterministic linking methods—such as Stata's merge or duplicates command—to fail. For example, one might need to link files that include only inconsistently spelled names, dates of birth with typos or missing data, and addresses that change over time. Probabilistic linkage methods score each potential pair of records on the probability the two records match, so that pairs with higher overall scores indicate a better match than pairs with lower scores. Two user-written Stata commands for probabilistic linking exist (`reclink` and `reclink2`), but they do not scale efficiently. `dtalink` is a new program that offers streamlined probabilistic linking methods implemented in parallelized Mata code. Significant improvements in speed make it practical to implement probabilistic linking methods on large, administrative data files (files with many rows or matching variables), and new features offer more flexible scoring and many-to-many matching techniques. This presentation introduces `dtalink`, discusses useful tips and tricks, and presents an example of linking data from Medicaid and birth certificates.

Background

- Stata users often need to:
  - Link records from two (or more) data files
  - Find duplicates within data files
- These two (related) problems date back to the very first computers
  - `merge-sort` algorithm invented by von Neumann in 1945 (Knuth 1987)
  - Precursors in the days before computers
- The era of "big data"
  - Ubiquitous applications bringing together data from disparate data systems to highly useful/informative ends

Data Linking

- Bring together separate pieces of information concerning a particular case
  - A case could be a person, a family, an event, a business, a location, or something else
  - Two (or more) input data files have one linking variable (or more) in common
- Match each case in File A with the corresponding case in File B
  - Final data stored in “long” or “wide” format (see `reshape`)
  - Essentially assigns a case identification (ID) number
- Goal
  - Identify all the cases (high sensitivity)
  - Do not inadvertently link two separate cases (high specificity)

Example: Before Linking

File A

	firstname	lastname	dob	ssn
1	Jane	Johnson	05/05/1985	1000000005
2	Amy	Miller	11/25/1985	1000000006
3	Mary	Smith	02/08/1985	1000000001
4	Amy	Miller	08/05/2000	1000000007
5	Elizabeth	Jones	05/05/1985	1000000003
6	Catherine	Johnson	05/05/1985	1000000002
7	Maria	Sanchez	01/01/1983	
8	Jane	Doe	01/05/1985	1000000000

File B

	Firstname	lastname	dob	ssn
1	Jane	Doe	01/06/1985	1000000000
2	Mary	Smoth	02/07/1985	
3	Katie	Jonson	05/05/1985	1000000002
4		Jones	05/05/1985	1000000003
5	Maria	Sanchez-Martinez	01/01/1983	1000000004
6	Jane	Johnson	05/05/1985	1000000005
7	Anne	Miller	05/01/1980	2000000007

Example: After Linking (Long Format)

	_matchID	fileid	firstname	lastname	dob	Ssn
Matched	1	A	Jane	Doe	01/05/1985	1000000000
	1	B	Jane	Doe	01/06/1985	1000000000
	2	A	Mary	Smith	02/07/1985	1000000001
	2	B	Mary	Smoth	02/08/1985	
	3	A	Catherine	Johnson	05/05/1985	1000000002
	3	B	Katie	Jonson	05/05/1985	1000000002
	4	A	Elizabeth	Jones	05/05/1985	1000000003
	4	B		Jones	05/05/1985	1000000003
	5	A	Maria	Sanchez	01/01/1983	
	5	B	Maria	Sanchez-Martinez	01/01/1983	1000000004
	6	A	Jane	Johnson	05/05/1985	1000000005
	6	B	Jane	Johnson	05/05/1985	1000000005
Not matched		A	Amy	Miller	08/05/2000	1000000007
		A	Amy	Miller	11/25/1985	1000000006
		B	Anne	Miller	05/01/1980	2000000007

Deduplication

- Identify multiple instances of the same case in a data file
  - One input file (instead of two)
- Highly related to data linking
  - "Linking a data file to itself"
  - Prevent row *i* from being linked to row *i*

Example: Before Deduplicating

firstname	lastname	dob	ssn
Jane	Doe	01/05/1985	1000000000
Mary	Smith	02/08/1985	1000000001
Catherine	Johnson	05/05/1985	1000000002
Katie	Jonson	05/05/1985	1000000002
Elizabeth	Jones	05/05/1985	1000000003
Maria	Sanchez	01/01/1983	
Jane	Johnson	05/05/1985	1000000005
Amy	Miller	08/05/2000	1000000007
Amy	Miller	11/25/1985	1000000006
Anne	Miller	05/01/1980	2000000007
Jane	Doe	01/06/1985	1000000000
Maria	Sanchez-Martinez	01/01/1983	1000000004
Mary	Smoth	02/07/1985	
	Jones	05/05/1985	1000000003

Example: After Deduplicating

	_matchID	firstname	lastname	dob	ssn
Deduplicated	1	Jane	Doe	01/05/1985	1000000000
	1	Jane	Doe	01/06/1985	1000000000
	2	Mary	Smith	02/08/1985	1000000001
	2	Mary	Smoth	02/07/1985	
	3	Catherine	Johnson	05/05/1985	1000000002
	3	Katie	Jonson	05/05/1985	1000000002
	4	Elizabeth	Jones	05/05/1985	1000000003
	4		Jones	05/05/1985	1000000003
	5	Maria	Sanchez	01/01/1983	
	5	Maria	Sanchez-Martinez	01/01/1983	1000000004
	6	Jane	Johnson	05/05/1985	1000000005
	6	Jane	Johnson	05/05/1985	1000000005
No duplicates found		Amy	Miller	08/05/2000	1000000007
		Amy	Miller	11/25/1985	1000000006
		Anne	Miller	05/01/1980	2000000007

Stata's Built-In Commands

- **Stata has built-in commands for data linking and deduplication**
  - **The `by:` prefix, `duplicates`, `merge`, and `joinby`**
- **Straightforward and efficient**
- **Only works for data file(s) with reliable or unique identifiers**
- **Need an approach for the (common) situation where the file(s) do not contain reliable or unique identifiers**
  - **Example:**
    - Names that are inconsistently spelled
    - Dates of birth with typos or missing data
    - Addresses and phone numbers that change over time

Deterministic Linking Methods

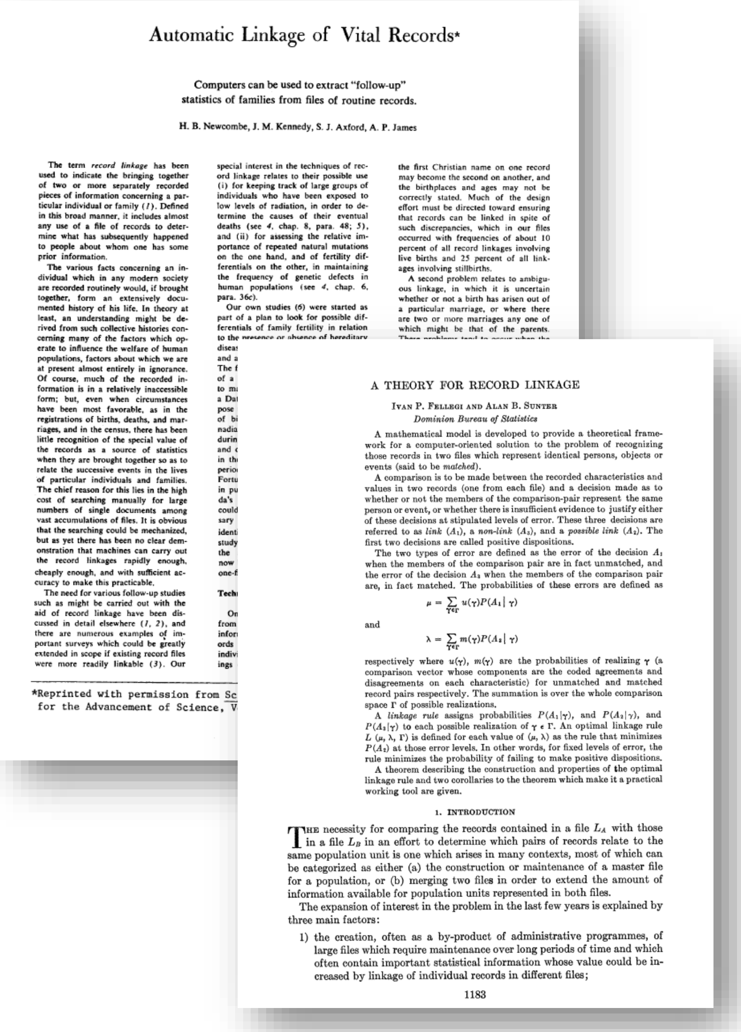
- **Conducted in multiple rounds**
  - **In the first round**
    - Linking is conducted using all (or most of) the linking variables.
    - If two records have the same data elements for all the linking variables, the records are identified
    - Matched records are set aside
  - **The next round begins**
    - Include all remaining records
    - Use a different combination of linking variables
  - **Weaker matching criteria are used in each round**
- **Match quality depends on the number of rounds and variables used in each round**

Pros and Cons of Deterministic Linking

- **Pros**
  - **Easy to understand and conduct**
  - **Uses Stata’s built-in commands**
  - **Valid and fast if direct identifiers are available**
- **Cons**
  - **Analyst arbitrarily sorts matching criteria from “strongest” to “weakest”**
  - **Typically developed through naive trial and error**
    - Labor intensive
    - Prone to mistakes
  - **Order of the rounds affects which records get matched**
    - Analyst must try different orderings
    - This rarely happens in practice
  - **False matches receive little attention**
  - **Explicit rules needed to handle missing data**

Probabilistic Linking Methods

- Ideas are around 60 years old
  - Newcombe et al. 1959
  - Fellegi and Sunter 1969
- Typically applied when there is no common unique identifier
- Core approach
  - Consider linking every potential pair of records
  - Compare matching variables for each potential pair
  - Score each potential pair of records on the probability that the two records match
  - Pairs with higher scores have higher probabilities of being a true match than do pairs with lower scores
    - Keep matches with high scores
    - Ignore the matches with low scores
    - Matches with scores in the middle can be manually reviewed
- Various extensions to this core approach
- Multiple software implementations
- Naturally handles missing data



Automatic Linkage of Vital Records*

Computers can be used to extract "follow-up" statistics of families from files of routine records.

H. B. Newcombe, J. M. Kennedy, S. J. Axford, A. P. James

The term *record linkage* has been used to indicate the bringing together of two or more separately recorded pieces of information concerning a particular individual or family (*1*). Defined in this broad manner, it includes almost any use of a file of records to determine what has subsequently happened to people about whom one has some prior information.

The various facts concerning an individual which in any modern society are recorded routinely would, if brought together, form an extensively documented history of his life. In theory at least, an understanding might be derived from such collective histories concerning many of the factors which operate to influence the welfare of human populations, factors about which we are at present almost entirely in ignorance. Of course, much of the recorded information is in a relatively inaccessible form; but, even when circumstances have been most favorable, as in the registrations of births, deaths, and marriages, and in the census, there has been little recognition of the special value of the records as a source of statistics when they are brought together so as to relate the successive events in the lives of particular individuals and families. The chief reason for this lies in the high cost of searching manually for large numbers of single documents among vast accumulations of files. It is obvious that the searching could be mechanized, but as yet there has been no clear demonstration that machines can carry out the record linkages rapidly enough, cheaply enough, and with sufficient accuracy to make this practicable.

The need for various follow-up studies such as might be carried out with the aid of record linkage have been discussed in detail elsewhere (*1*, *2*), and there are numerous examples of important surveys which could be greatly extended in scope if existing record files were more readily linkable (*3*). Our special interest in the techniques of record linkage relates to their possible use (i) for keeping track of large groups of individuals who have been exposed to low levels of radiation, in order to determine the causes of their eventual deaths (see *4*, chap. 8, para. 48; *5*), and (ii) for assessing the relative importance of repeated natural mutations on the one hand, and of fertility differentials on the other, in maintaining the frequency of genetic defects in human populations (see *4*, chap. 6, para. 36*c*).

Our own studies (*6*) were started as part of a plan to look for possible differentials of family fertility in relation to the presence or absence of hereditary

the first Christian name on one record may become the second on another, and the birthplaces and ages may not be correctly stated. Much of the design effort must be directed toward ensuring that records can be linked in spite of such discrepancies, which in our files occurred with frequencies of about 10 percent of all record linkages involving live births and 25 percent of all linkages involving stillbirths.

A second problem relates to ambiguous linkage, in which it is uncertain whether or not a birth has arisen out of a particular marriage, or where there are two or more marriages any one of which might be that of the parents.

*Reprinted with permission from Sc
for the Advancement of Science, V

A THEORY FOR RECORD LINKAGE

Ivan P. Fellegi and Alan B. Sunter

Dominion Bureau of Statistics

A mathematical model is developed to provide a theoretical framework for a computer-oriented solution to the problem of recognizing those records in two files which represent identical persons, objects or events (said to be *matched*).

A comparison is to be made between the recorded characteristics and values in two records (one from each file) and a decision made as to whether or not the members of the comparison-pair represent the same person or event, or whether there is insufficient evidence to justify either of these decisions at stipulated levels of error. These three decisions are referred to as *link* (A_1), a *non-link* (A_3), and a *possible link* (A_2). The first two decisions are called positive dispositions.

The two types of error are defined as the error of the decision A_1 when the members of the comparison pair are in fact unmatched, and the error of the decision A_3 when the members of the comparison pair are, in fact matched. The probabilities of these errors are defined as

$$\mu = \sum_{\gamma \epsilon \Gamma} u(\gamma) P(A_1 | \gamma)$$

and

$$\lambda = \sum_{\gamma \epsilon \Gamma} m(\gamma) P(A_3 | \gamma)$$

respectively where $u(\gamma)$, $m(\gamma)$ are the probabilities of realizing γ (a comparison vector whose components are the coded agreements and disagreements on each characteristic) for unmatched and matched record pairs respectively. The summation is over the whole comparison space Γ of possible realizations.

A *linkage rule* assigns probabilities $P(A_1|\gamma)$, and $P(A_2|\gamma)$, and $P(A_3|\gamma)$ to each possible realization of $\gamma \epsilon \Gamma$. An optimal linkage rule L (μ, λ, Γ) is defined for each value of (μ, λ) as the rule that minimizes $P(A_2)$ at those error levels. In other words, for fixed levels of error, the rule minimizes the probability of failing to make positive dispositions.

A theorem describing the construction and properties of the optimal linkage rule and two corollaries to the theorem which make it a practical working tool are given.

1. INTRODUCTION

The necessity for comparing the records contained in a file L_A with those in a file L_B in an effort to determine which pairs of records relate to the same population unit is one which arises in many contexts, most of which can be categorized as either (a) the construction or maintenance of a master file for a population, or (b) merging two files in order to extend the amount of information available for population units represented in both files.

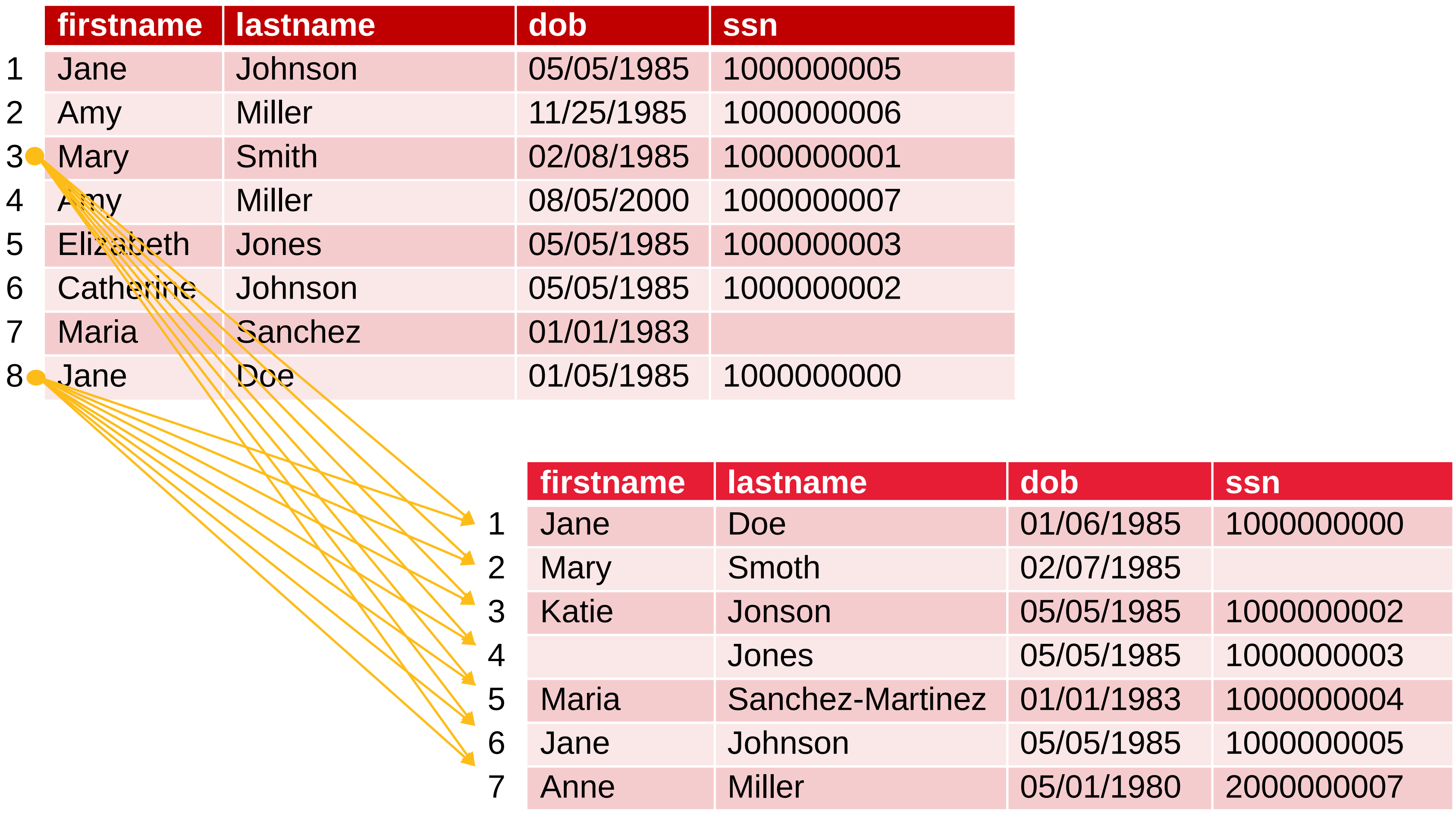
The expansion of interest in the problem in the last few years is explained by three main factors:

1) the creation, often as a by-product of administrative programmes, of large files which require maintenance over long periods of time and which often contain important statistical information whose value could be increased by linkage of individual records in different files;

1183

Example: Probabilistic Data Linking (1)

- Every row in file A is compared to every row in file B



	firstname	lastname	dob	ssn
1	Jane	Johnson	05/05/1985	1000000005
2	Amy	Miller	11/25/1985	1000000006
3	Mary	Smith	02/08/1985	1000000001
4	Amy	Miller	08/05/2000	1000000007
5	Elizabeth	Jones	05/05/1985	1000000003
6	Catherine	Johnson	05/05/1985	1000000002
7	Maria	Sanchez	01/01/1983	
8	Jane	Doe	01/05/1985	1000000000

	firstname	lastname	dob	ssn
1	Jane	Doe	01/06/1985	1000000000
2	Mary	Smoth	02/07/1985	
3	Katie	Jonson	05/05/1985	1000000002
4		Jones	05/05/1985	1000000003
5	Maria	Sanchez-Martinez	01/01/1983	1000000004
6	Jane	Johnson	05/05/1985	1000000005
7	Anne	Miller	05/01/1980	2000000007

Example: Probabilistic Data Linking (2)

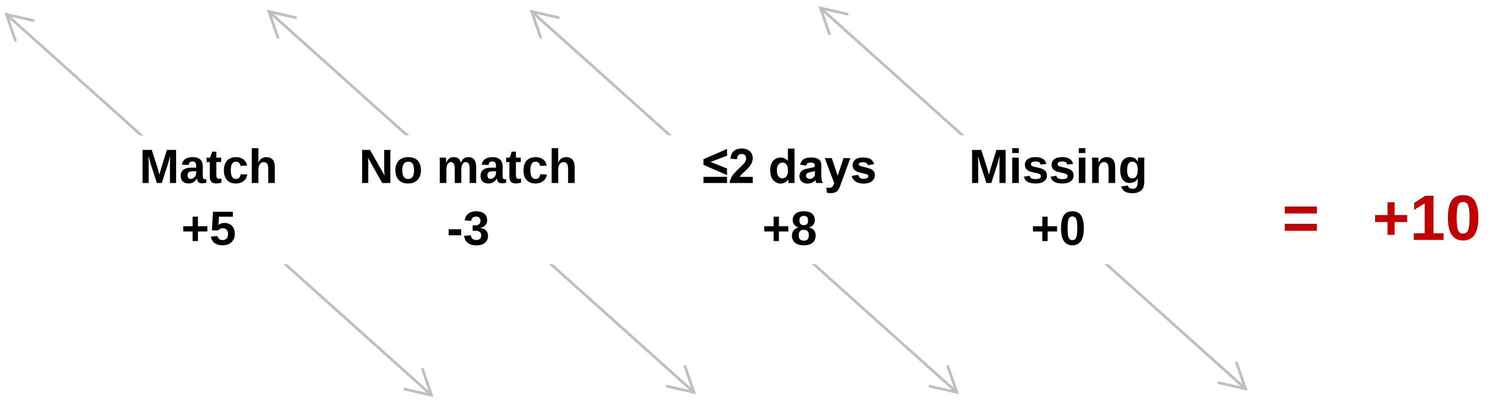
- User chooses matching variables and calipers
- Every matching variable is assigned positive and negative weights for matches and non-matches, respectively

	firstname	lastname	dob	ssn
Weight applied if variable matches	+5	+7	+8	+15
Weight applied if variable does not match	-3	-3	-2	-5
Weight applied if data are missing	0	0	0	0
Definition of a "match"	Exact match	Exact match	Within 2 days	Exact match

Example: Probabilistic Data Linking (3)

- Each potential pair is scored using sum of the weights
- Pairs with higher overall scores indicate a better match than pairs with lower scores

	firstname	lastname	dob	ssn
3	Mary	Smith	02/08/1985	1000000001



Match
+5

No match
-3

≤2 days
+8

Missing
+0

= +10

	firstname	lastname	dob	ssn
2	Mary	Smoth	02/07/1985	

Example: Probabilistic Data Linking (4)

- **Compare scores across all potential matched pairs**

	firstname	lastname	dob	ssn
1	Jane	Johnson	05/05/1985	1000000005
2	Amy	Miller	11/25/1985	1000000006
3	Mary	Smith	02/08/1985	1000000001
4	Amy	Miller	08/05/2000	1000000007
5	Elizabeth	Jones	05/05/1985	1000000003
6	Catherine	Johnson	05/05/1985	1000000002
7	Maria	Sanchez	01/01/1983	
8	Jane	Doe	01/05/1985	1000000000

Score		firstname	lastname	dob	ssn
-13	1	Jane	Doe	01/06/1985	1000000000
+10	2	Mary	Smoth	02/07/1985	
-13	3	Katie	Jonson	05/05/1985	1000000002
-10	4		Jones	05/05/1985	1000000003
-13	5	Maria	Sanchez-Martinez	01/01/1983	1000000004
-13	6	Jane	Johnson	05/05/1985	1000000005
-13	7	Anne	Miller	05/01/1980	2000000007

Example: Probabilistic Data Linking (5)

- **Keep matched pairs with high scores, assign pair IDs**

	_matchID	_score	fileid	firstname	lastname	dob	Ssn
Matched	1	25.00	A	Jane	Doe	01/05/1985	1000000000
	1	25.00	B	Jane	Doe	01/06/1985	1000000000
	2	10.00	A	Mary	Smith	02/08/1985	1000000001
	2	10.00	B	Mary	Smoth	02/07/1985	
	3	17.00	A	Catherine	Johnson	05/05/1985	1000000002
	3	17.00	B	Katie	Jonson	05/05/1985	1000000002
	4	30.00	A	Elizabeth	Jones	05/05/1985	1000000003
	4	30.00	B		Jones	05/05/1985	1000000003
	5	10.00	A	Maria	Sanchez	01/01/1983	
	5	10.00	B	Maria	Sanchez-Martinez	01/01/1983	1000000004
	6	35.00	A	Jane	Johnson	05/05/1985	1000000005
	6	35.00	B	Jane	Johnson	05/05/1985	1000000005
Not matched			A	Amy	Miller	08/05/2000	1000000007
			A	Amy	Miller	11/25/1985	1000000006
			B	Anne	Miller	05/01/1980	2000000007

`dtalink`
A New Package for Stata

Other Probabilistic Linking Packages

- **Stata**
  - `reclink` **and** `reclink2`
    - Do not scale efficiently (not well-parallelized)
    - Crash with many matching variables
    - Embedded features (for example, complex string comparisons) add to runtimes or have limited flexibility (for example, specification of blocking variables)
- **Point-and-click software**
  - **Link Plus, LinkageWIZ, and Link King**
    - Not programmable
    - Strict limits on file sizes (rows and/or columns)
    - Required variables or features specific to certain variable types (for example, names)
    - Special features add to runtimes (for example, string comparisons, specification of blocking variables)
- **Other languages**
  - `RecordLinkage` **(R)**
    - Sophisticated, but stores all potential pairs in memory
  - `FastLink` **(R)**
  - **FEBRL and Python Record Linkage Toolkit (Python)**

Goals for `dtalink` and Approach Taken

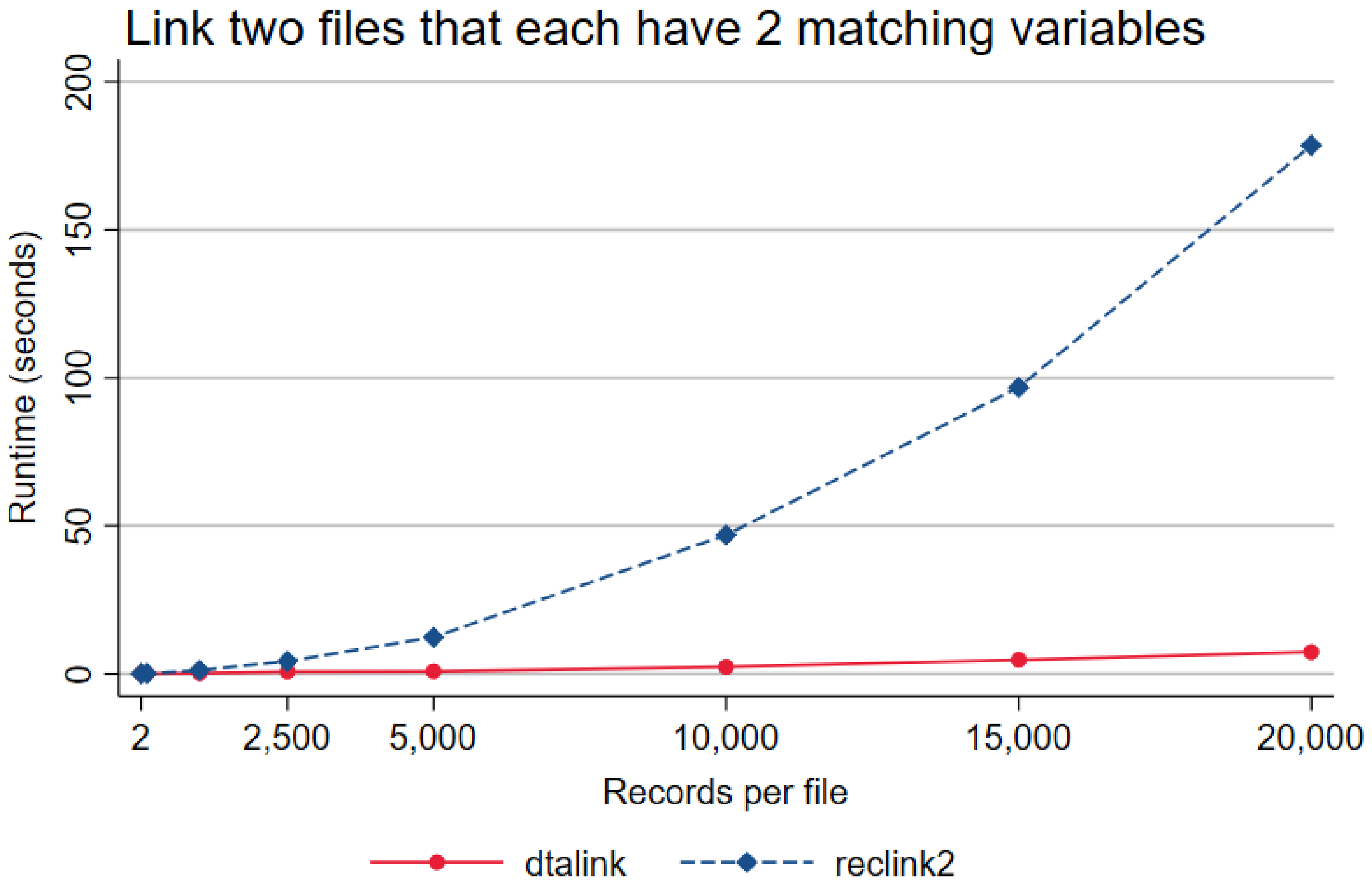
Goals

- **Stata implementation of probabilistic matching methods**
  - Data linking
  - Deduplication
- **Generic:** Works with any data file
- **Fast, even with large data files**
- **Two distance measures**
  - Exact matching
  - Caliper matching
- **Several new pre- and post-processing options**
  - Many-to-many matching techniques
  - Multiple rows per case
  - Estimate matching weights

Programming approach

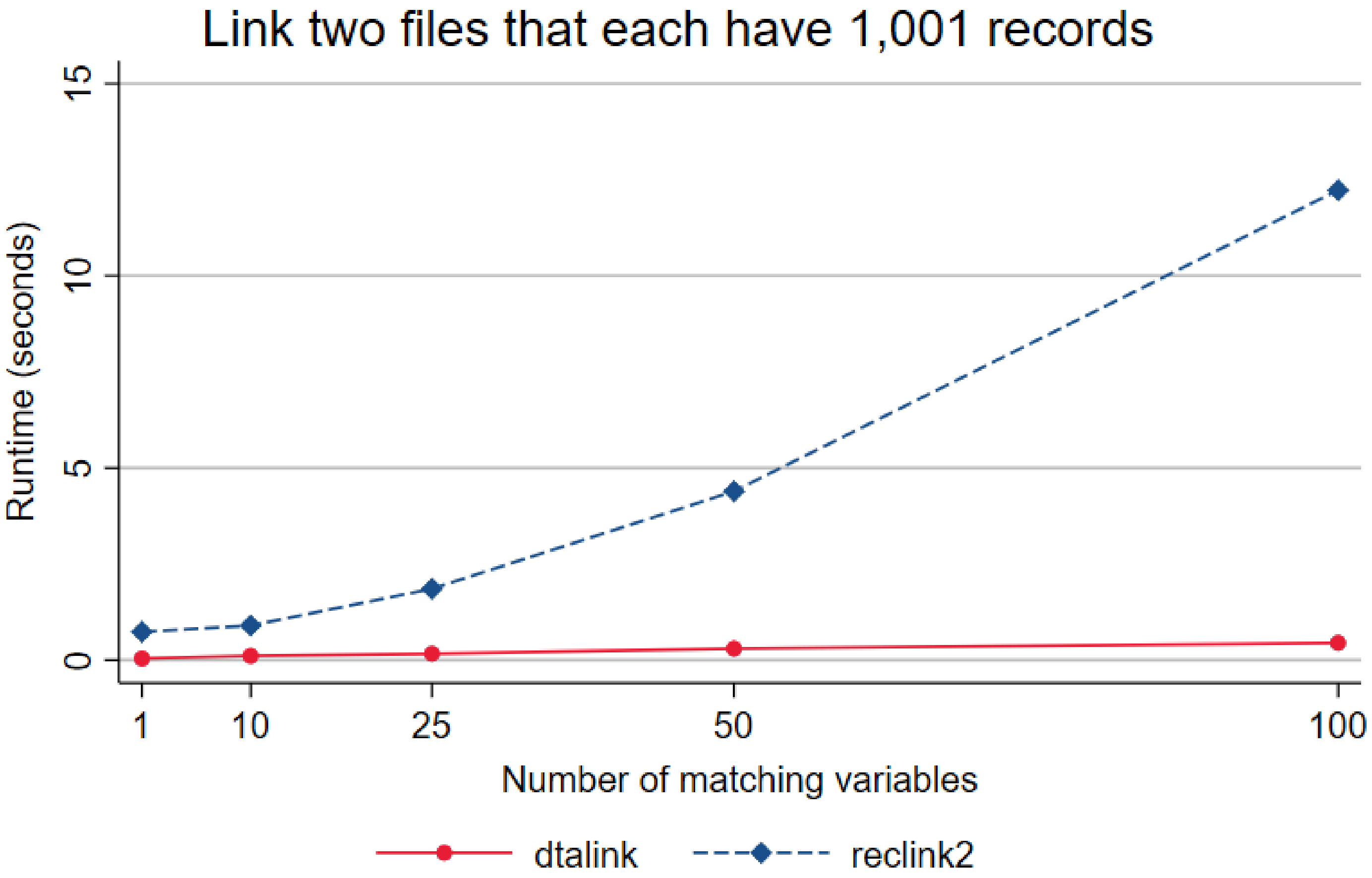
- **Object-oriented programming (class) in Mata**
  - Stata (.ado) wrapper command
- **Memory efficient “hot” loops**
  - Data are not copied inside class functions
  - Non-matches are not stored
- **Highly parallelized**
  - Mata colon operators
- **“No frills”**
  - No unnecessary computations in program’s kernel

dtalink Is Faster with Many Records



Note: Simulation used four processors and random data.

dtalink Is Faster with Many Matching Variables



Note: reclink2 crashed with 500 or 1,000 variables; dtalink required 5.1 and 10.4 seconds to link files with 500 and 1,000 variables, respectively. Simulation used four processors and random data.

Syntax

Deduplication:

```
. dtalink dspecs [if] [in] [, options]
```

Data linking:

```
. dtalink dspecs [if] [in] using filename [, options]
. dtalink dspecs [if] [in], source(varname) [options]
```

where

- *dspecs* = *dspec* [*dspec* [*dspec* [...]]]
- *dspec* = *varname* #1 #2 [#3]
- *varname* **is a matching variable**
- #1 **is the weight applied for a match (positive)**
- #2 **is the weight applied for a non-match (negative)**
- #3 **is the caliper for distance matching (≥ 0)**
  - #3 is optional and is only allowed with numeric variables
  - Exact matching is used if #3 is not specified)

Examples from Above

- Deduplication

```
. dtalink firstname 5 -3 lastname 7 -3 dob 8 -2 ssn 15 -5, ///
          cutoff(3)
```

- Data linking (syntax 1)

```
. dtalink firstname 5 -3 lastname 7 -3 dob 8 -2 ssn 15 -5 ///
          using myfilename.dta, cutoff(3)
```

- Data linking (syntax 2)

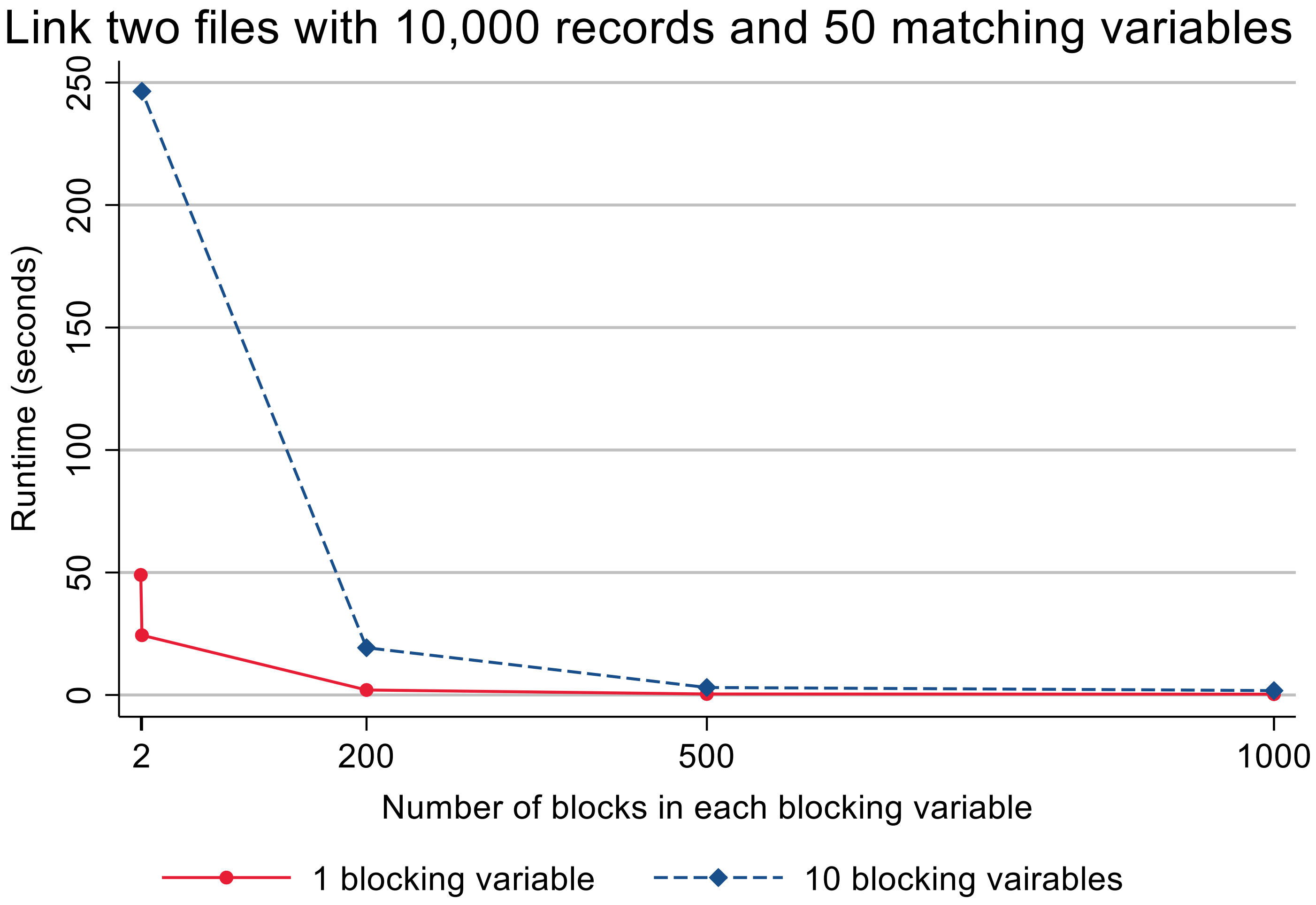
```
. append using myfilename.dta, gen(sourcevar)

. dtalink firstname 5 -3 lastname 7 -3 dob 8 -2 ssn 15 -5, ///
          cutoff(3) source(sourcevar)
```

Key Options

- `cutoff(#)`
  - Specifies the minimum score required to keep a potential matched pair
- `id(`*`varname`*`)`
  - Provides a variable to identify unique case
  - Each case may have more than one record (row)
    - Cases scored using the two best-matching records
  - Example: Administrative files with more than one row per person
- `block(`*`blocklist`*`)`
  - Limits comparisons to records that match on at least one "blocking variable"
    - Blocking variables can be matching variables, and vice versa
  - Significantly reduces runtimes
    - Without blocking, probabilistic matching can be computationally intensive (billions of pairs assessed)
  - Users can avoid missed matched pairs by blocking data multiple times using different variables (or combinations of variables)

Linking Is Faster with Blocking



Additional Options

- `bestmatch and srcbestmatch(0|1)`
  - **Enables 1:1, 1:M, or M:1 matching**
  - **Example:**
    - Match individuals' records across two files (1:1 matching)
    - Match children in one file to mothers in a second file (M:1 matching)
- `combinesets`
  - **Combines matched sets**
  - **Most useful when deduplicating files**
    - For example, when one person appears three or more times
- `calcweights`
  - **Tracks the percentage of times a variable matches, separately for potential match pairs above and below the cutoff**
  - **Uses these percentages to compute recommended weights (for the next run)**
  - **Increases runtimes**

Tips and Tricks for Using `dtalink`

Tip #1: Preprocessing Data

- The most important step is preparing data for linkage (Herzog et al. 2007)
- Easy data cleaning steps can greatly improve results
  - Convert strings to upper (or lower) case
    - See `. help string functions` in Stata
  - Split dates and addresses into subcomponents
    - See `. help datetime functions` in Stata
    - `stnd_address` (`reclink2` package)
  - Standardize abbreviations and other values
    - “Street” versus “St.”, “Ohio” versus “OH”
    - `stnd_*` commands (`reclink2` package)
  - Remove or standardize punctuation
    - `regexr()` function in Stata
  - Use phonetic algorithms to code names
    - `soundex()` function in Stata, `nysiis` package

Example: Phonetic Algorithms

```
. nysiis name, generate(name_nysiis)
. dtalink name 7 0 name_nysiis 3 -2
```

+10 if records have the same name (+7)
and therefore have the same NYIIS code (+3)

+3 if records have the same NYIIS code (+3)
but don't have the same name (0)

-2 if records don't have the same NYIIS code (-2)
and therefore don't have the same name (0)

Tip #2: Getting Weights and Cutoffs

- **Weights**
  - **Reflect the probability a variable matches for true versus false matches**
    - Higher for linking variables with more specificity (like SSN)
    - Lower for variables with less specificity (like city or age)
  - **Estimate using a training data file and the `calcweights` option**
    - Without training data, consider using a good linking variable (like SSN) and the `calcweights` option to obtain recommended weights for the remaining matching variables
  - **Could estimate using advanced predictive analytic techniques**
- **Cutoffs**
  - **Consider hypothetical cases**
    - "What if records match on SSN and date of birth, but nothing else?"
  - **Start with a low cutoff, review, and delete pairs if a higher cutoff is needed**
- **With well-calibrated weights, a good cutoff would have only a few matched pairs near it, with a mix of "good" and "bad" matches**
- **No universal best approach**
  - **Users need to weigh inherent trade-offs between sensitivity and specificity**

Tip #3: Vary Weights for Rare Values or Closer Matches

- Advanced methods
  - Assign partial weight in cases where a variable matches closely, but not exactly
  - “Bonus” weights when multiple variables match
  - Assign partial weight for strings that are slightly different (Winkler 1990)
  - Account for relative frequencies within a variable (Jaro 1995)
    - “Kranker” versus “Smith”
    - Small versus large ZIP codes
- These techniques were not built into dtalink directly
  - Not all users would have wanted the feature, given the increased computational burden
- But similar results can be achieved with clever use of available options
  - See examples in `dtalink` documentation

Example: Weights for a Matching Variable Vary

```
. bysort name: gen name_count = _N
. clonevar rare_name = name if name_count<=50
. dtalink name 7 -5 rare_name 3 -2
```

+10/-7 for matches/nonmatches on "rare" names (< 50 rows in file)

+7/-5 for matches/nonmatches on "common" names

```
. dtalink date 5 0 date 3 0 7 date 2 -6 30
```

+10 if date matches exactly (5+3+2)

+5 if date matches within 1 to 7 days (0+3+2)

+2 if date matches within 8 to 30 days (0+0+2)

-6 if date does not match within 30 days (0+0-2)

```
. dtalink date 5 0 month 3 0 year 2 -6
```

+10 if date matches exactly (5+3+2)

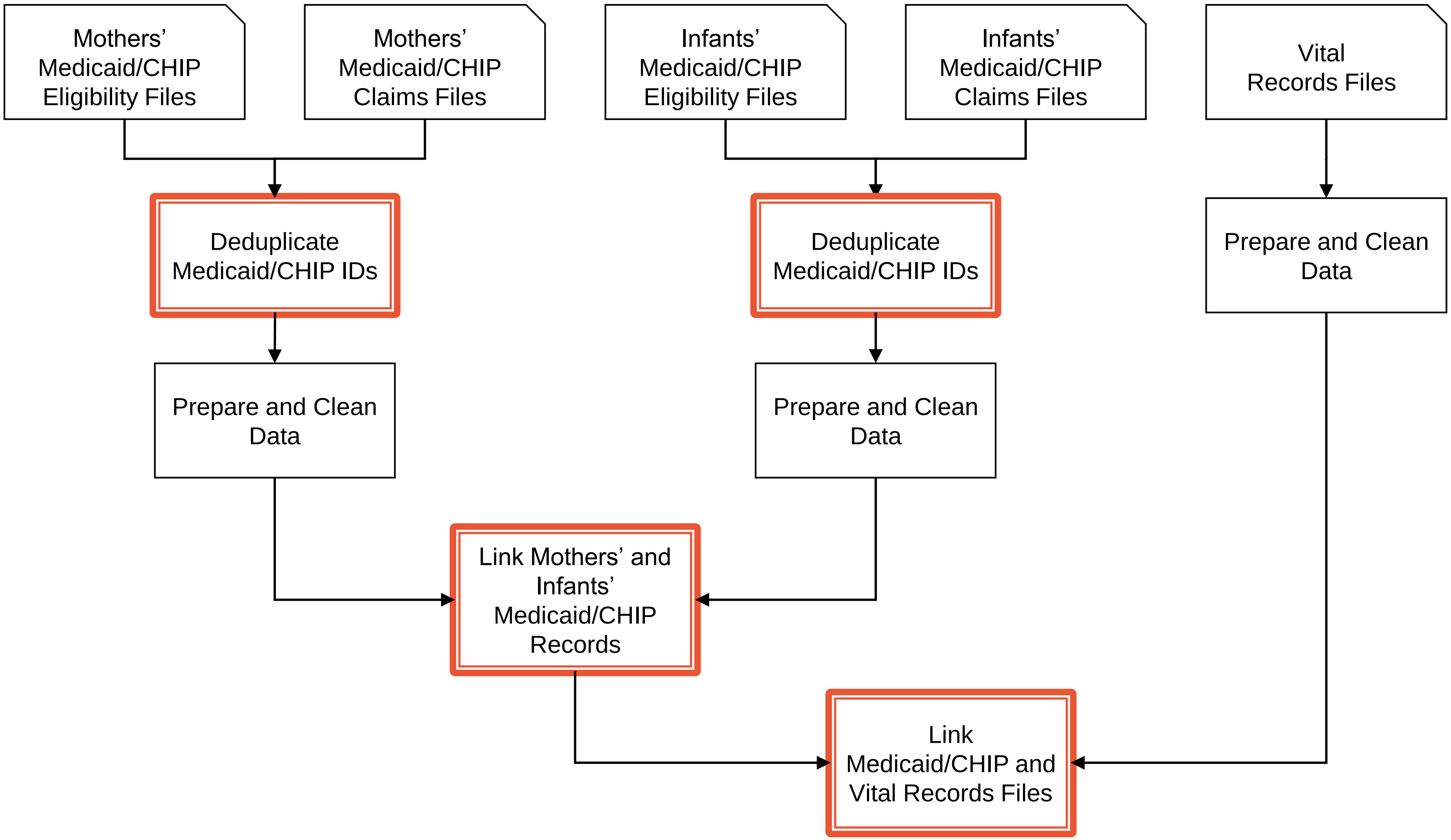
+5 if date month and year match, but not day (0+3+2)

+2 if year matches, but not month or day (0+0+2)

-6 if date does not match (0+0-6)

Linking Medicaid/CHIP Administrative Data with Vital Records Data (Birth Certificates) for Mothers and Infants

Overview of the Process



Mothers' Medicaid/CHIP Eligibility Files

Mothers' Medicaid/CHIP Claims Files

Infants' Medicaid/CHIP Eligibility Files

Infants' Medicaid/CHIP Claims Files

Vital Records Files

Deduplicate Medicaid/CHIP IDs

Deduplicate Medicaid/CHIP IDs

Prepare and Clean Data

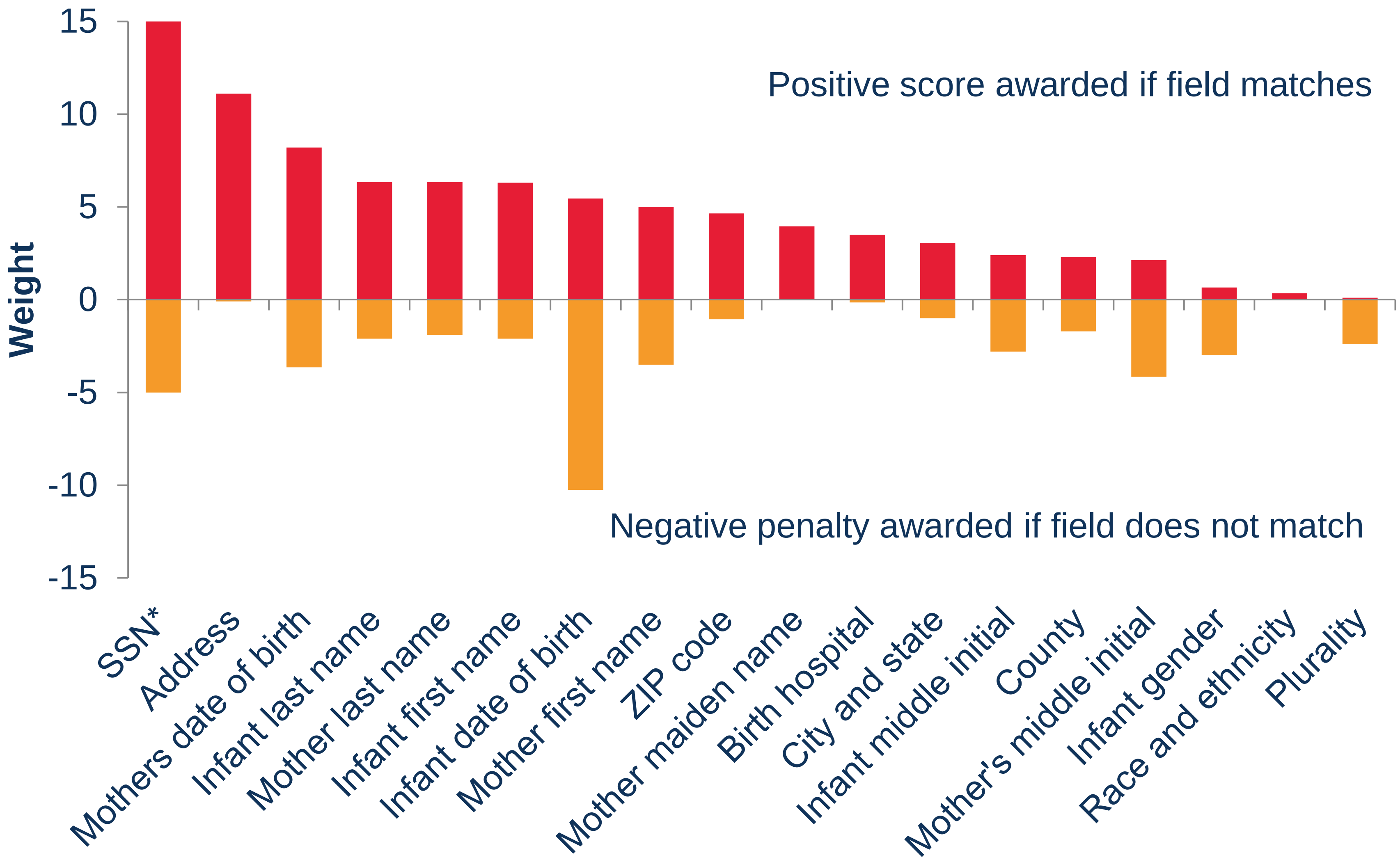
Prepare and Clean Data

Prepare and Clean Data

Link Mothers' and Infants' Medicaid/CHIP Records

Link Medicaid/CHIP and Vital Records Files

Illustrative Matching Variables and Weights



Scale of Linking Medicaid to Vital Records Data in One Medium-Sized State

- Matching mother-infant dyads to vital records
  - 360,000 birth certificates
  - 1.6 million records in Medicaid files
  - 4 dozen matching variables
  - Multiple blocking variables with large blocks
  - Approximately 15 hours to run
- Repeat with unmatched mothers and unmatched infants
  - Smaller files
  - Create new mother-infant dyads when a mother and infant are linked to the same birth certificate

References

Fellegi, I. P., and A. B. Sunter. 1969. “A Theory for Record Linkage.” *Journal of the American Statistical Association*, vol. 64, no. 328, 1969, p. 1183. doi:10.2307/2286061.

Herzog, T. N., F. J. Scheuren, and W. E. Winkler. *Data Quality and Record Linkage Techniques*. New York: Springer, 2007.

Jaro, M. A. “Probabilistic Linkage of Large Public Health Data Files.” *Statistics in Medicine*, vol. 14, no. 5–7, 1995, pp. 491–498.

Knuth, D. E. “Von Neumann’s First Computer Program.” In *Papers of John von Neumann on Computing and Computer Theory*, edited by W. Aspray and A. Burks. Cambridge: MIT Press, 1987.

Newcombe, H. B., J. M. Kennedy, S. J. Axford, and A. P. James. “Automatic Linkage of Vital Records.” *Science*, vol. 130, no. 3381, October 1959, pp. 954–959.

Winkler, W. E. “String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage.” Pages 354–359 in *1990 Proceedings of the Section on Survey Research*. Alexandria, VA: American Statistical Association, 1990.