Data Integration

Goal :To identify the same real world entity in different tables

Other names:

- Record Linkage
- Entity Resolution
- Deduplication (Link to self)
- Merge / Purge

Hye-Chung Kum

Population Informatics Research Group

http://research.tamhsc.edu/pinformatics/

http://pinformatics.web.unc.edu/

Course URL:

http://pinformatics.tamhsc.edu/phpm672

Assign 3

- Average: 5.6 (no 8)
- Average so far
  - Two groups
- Issues
  - None or incorrect readme (-1)
  - Incorrect or no while loop: (2/4)
  - No descriptive analysis
  - tot2010= sum (of dc20101-dc20104);
  - Missing files v1-v3
- Folder location
  - Pwd
- Review code

Record Linkage Example

EISID : E1	EISID : E2	EISID : E3	EISID : E4
ssn : 085-66-9980 first name : Sally last name : Hill MI : L DOB : 3/4/1999	ssn : 143-25-9304 first name : Emily last name : Brown MI : K DOB : 6/2/2004	ssn : 354-563-2343 first name : Mary last name : Johnson MI : G DOB : 5/13/1983	ssn : 532-34-9183 first name : David last name : Ford MI : J DOB : 10/2/1990

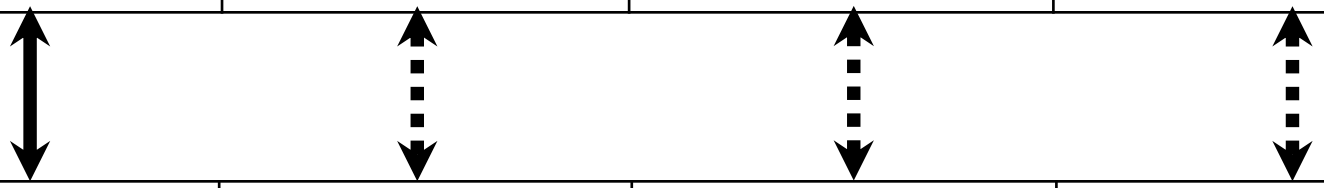
SISID : S1	SISID : S2	SISID : S3	SISID : S4
ssn : 085-66-9980 first name : Sally last name : Hill MI : L DOB : 3/4/1999	ssn : 143-52-9304 first name : Emily last name : Brown MI : K DOB : 6/2/2004	ssn : 354-563-2343 first name : Mary last name : Hawkins MI : J DOB : 5/13/1983	ssn : 532-34-9183 first name : David last name : Ford MI : J DOB : 2/10/1990

Inherent Nature of Real Data

- Data are expressed differently
  - nick names
- Data change over time
  - person's last name
- Data are not unique attributes
  - John Smith
- Missing Data
  - ssn are often missing
- Errors in Data
  - Rule of thumb : 5% error in administrative data

Record Linkage Example

EISID : E1	EISID : E2	EISID : E3	EISID : E4
ssn : 085-66-9980 first name : Sally last name : Hill MI : L DOB : 3/4/1999	ssn : 143-25-9304 first name : Emily last name : Brown MI : K DOB : 6/2/2004	ssn : 354-563-2343 first name : Mary last name : Johnson MI : G DOB : 5/13/1983	ssn : 532-34-9183 first name : David last name : Ford MI : J DOB : 10/2/1990



SISID : S1	SISID : S2	SISID : S3	SISID : S4
ssn : 085-66-9980 first name : Sally last name : Hill MI : L DOB : 3/4/1999	ssn : 143-52-9304 first name : Emily last name : Brown MI : K DOB : 6/2/2004	ssn : 354-563-2343 first name : Mary last name : Hawkins MI : J DOB : 5/13/1983	ssn : 532-34-9183 first name : David last name : Ford MI : J DOB : 2/10/1990

What does this mean?

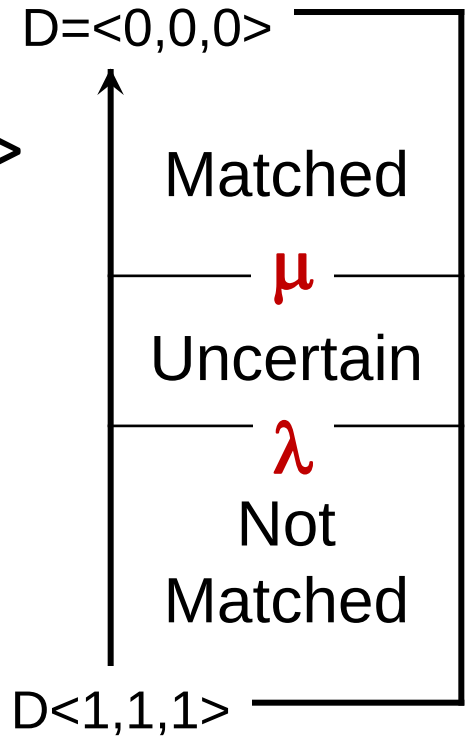
- Exact match will not work
  - Only 60% to 70% with exact match
  - Privacy protection through one way hash
  - Privacy preserving using set union
- Must have approximate match !
  - Probably will require some manual review of "uncertain region"

Approximate Matching Methods

- Capture as many of the false negatives
- While introducing as little of the false positives
- Probabilistic Methods
  - Naïve Bayes : Probabilistic Record Linkage
    - Newcombe (1959)
    - by Fellegi and Sunter (1969)
  - Other Machine learning models
    - Actively learning
- Deterministic Methods

Probabilistic Record Linkage

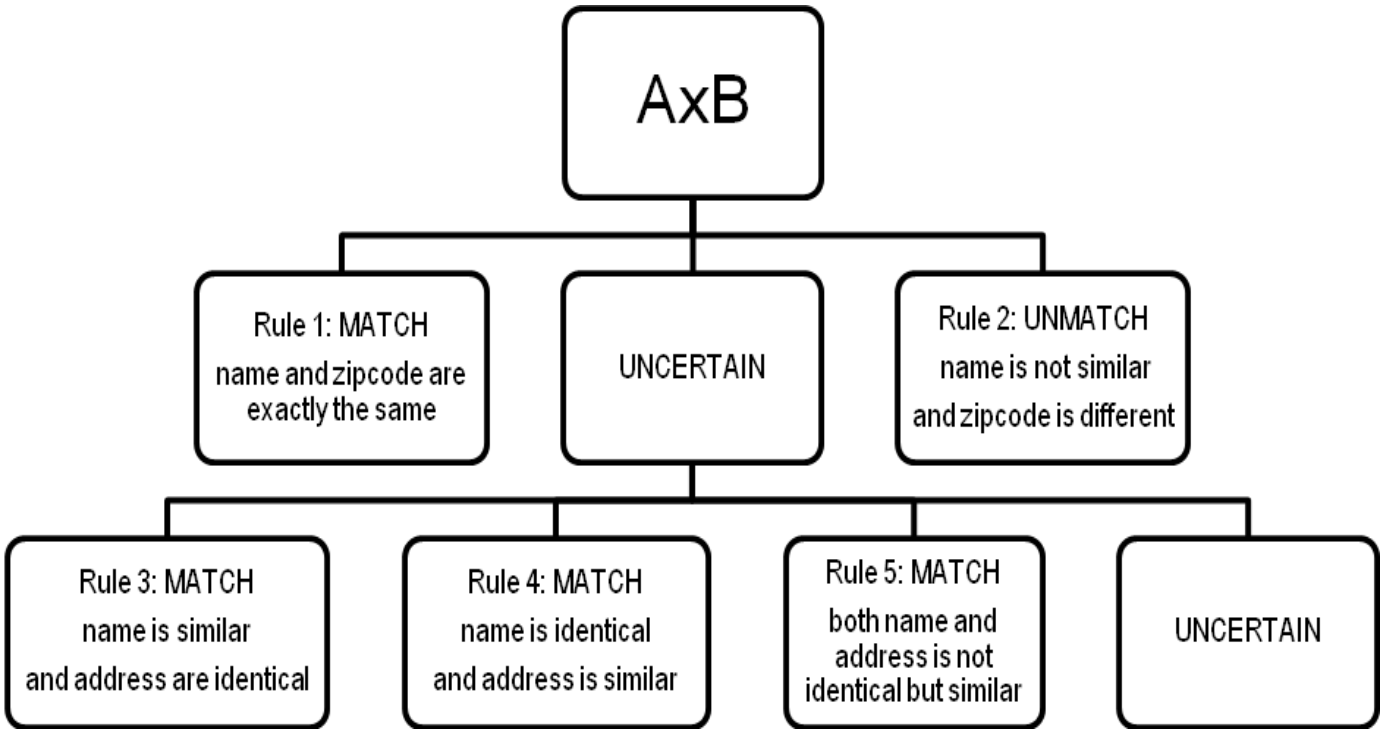
- Block/Score
- D = <$dist_{SSN}$, $dist_{NAME}$, $dist_{DOB}$ >
- Train model : Need test data
- Estimate the two threshold
- Resolve the uncertain region manually
- Naïve Bayes Model



$$(\mathbf{R_A}, \mathbf{R_B}) \in \begin{cases} M & if\ l(\underline{s}) = \dfrac{p(\underline{s}|M)}{p(\underline{s}|U)} \geq \dfrac{p(U)}{p(M)} \\ U & otherwize \end{cases}, \qquad where\ l(\underline{s}) = \frac{p(\underline{s}|M)}{p(\underline{s}|U)}\ is\ the\ likelihood\ ratio$$

Deterministic Matching Methods

- Rule Based : iterative



Comparison

- Exact Matching
  - Only when data is clean.
  - Great when it works, but doesn't work in many situations
  - Example: SSN, County FIPS Code
- Deterministic Approximate Matching
  - Easier to interpret/control
  - Can manage complexity to levels desired
  - More difficult to fine tune for complex data
- Probabilistic Approximate Matching
  - Can handle more complex data
  - Depends on the data being linked
  - Difficult to interpret what is being linked or not.

Example from papers

- SEER
  - Boscoe FP, Schrag D, Chen K, et al. Building capacity to assess cancer care in the Medicaid population in New York State. *Health Services Research* 2011;46(3):805-20.
- Vital records
  - Bronstein J, Lomatsch C, Fletcher D, Wooten T, Lin TM, Nugent R, Lowery C. Issues and Biases in Matching Medicaid Pregnancy Episodes to Vital Records Data: The Arkansas Experience. *Maternal and Child Health Journal*, 2009;13(2):250-259

For assignment 4

- Nothing complex
- But must do some sort of approximate linkage
- OR find the “different” data, and clean it

Cleaning Data Example

EISID : E1	EISID : E2	EISID : E3	EISID : E4
ssn : 085-66-9980 first name : Sally last name : Hill MI : L DOB : 3/4/1999	ssn : 143-25-9304 first name : Emily last name : Brown MI : K DOB : 6/2/2004	ssn : 354-563-2343 first name : Mary last name : Johnson MI : G DOB : 5/13/1983	ssn : 532-34-9183 first name : David last name : Ford MI : J DOB : 10/2/1990

```
* Note that you do not know which is correct;
* But you have to sync it to one value;
if ssn= '532-34-9183' then dob=mdy(10,2,1990);
```

ssn : 085-66-9980 first name : Sally last name : Hill MI : L DOB : 3/4/1999	ssn : 143-52-9304 first name : Emily last name : Brown MI : K DOB : 6/2/2004	ssn : 354-563-2343 first name : Mary last name : Hawkins MI : J DOB : 5/13/1983	ssn : 532-34-9183 first name : David last name : Ford MI : J DOB : 2/10/1990

Finding duplicate records

```
* Both tables are sorted by county;
* If need to find duplicates in multiple vars;
* Combine the multiple vars into one variable
first, then run same code;

data dupcnty;
merge tab1 tab2;
by county;
if !(first.county & last.county);
```

Approximate Matching Example standardize on caps

EISID : E1	EISID : E2	EISID : E3	EISID : E4
ssn : 085-66-9980 first name : Sally last name : Hill MI : L DOB : 3/4/1999	ssn : 143-25-9304 first name : Emily last name : Brown MI : K DOB : 6/2/2004	ssn : 354-563-2343 first name : Mary last name : Johnson MI : G DOB : 5/13/1983	ssn : 532-34-9183 first name : David last name : ford MI : J DOB : 10/2/1990

```
* Create a new standardize variable to link on;
linklname=lowcase(lname);
```

ssn : 085-66-9980 first name : Sally last name : Hill MI : L DOB : 3/4/1999	ssn : 143-52-9304 first name : Emily last name : Brown MI : K DOB : 6/2/2004	ssn : 354-563-2343 first name : Mary last name : Hawkins MI : J DOB : 5/13/1983	ssn : 532-34-9183 first name : David last name : Ford MI : J DOB : 2/10/1990

Approximate Matching Example standardize on space

EISID : E1	EISID : E2	EISID : E3	EISID : E4
ssn : 085-66-9980 first name : Sally last name : Hill MI : L DOB : 3/4/1999	ssn : 143-25-9304 first name : Emily last name : Brown MI : K DOB : 6/2/2004	ssn : 354-563-2343 first name : Mary last name : Johnson MI : G DOB : 5/13/1983	ssn : 532-34-9183 first name : David last name : fordJr MI : J DOB : 10/2/1990

```
* Create a new standardize variable to link on;
linklname=compress(lowcase(lname));
```

ssn : 085-66-9980 first name : Sally last name : Hill MI : L DOB : 3/4/1999	ssn : 143-52-9304 first name : Emily last name : Brown MI : K DOB : 6/2/2004	ssn : 354-563-2343 first name : Mary last name : Hawkins MI : J DOB : 5/13/1983	ssn : 532-34-9183 first name : David last name : Ford Jr MI : J DOB : 2/10/1990

Approximate Matching Example standardize on space

EISID : E1	EISID : E2	EISID : E3	EISID : E4
ssn : 085-66-9980 first name : Sally last name : Hill MI : L DOB : 3/4/1999	ssn : 143-25-9304 first name : Emily last name : Brown MI : K DOB : 6/2/2004	ssn : 354-563-2343 first name : Mary last name : Johnson MI : G DOB : 5/13/1983	ssn : 532-34-9183 first name : David last name : ford MI : J DOB : 10/2/1990

```
* Create a new standardize variable to link on;
linklname=compress(lowcase(lname);
linklname=tranwrd(linkname, 'jr', '' );
```

ssn : 085-66-9980 first name : Sally last name : Hill MI : L DOB : 3/4/1999	ssn : 143-52-9304 first name : Emily last name : Brown MI : K DOB : 6/2/2004	ssn : 354-563-2343 first name : Mary last name : Hawkins MI : J DOB : 5/13/1983	ssn : 532-34-9183 first name : David last name : Ford Jr MI : J DOB : 2/10/1990

Other useful functions

- Appendix 2 (p59) of ARHQ Report

```
vto=translate(vfrom,'          ',"()',-.");
vto=lowcase(compress(vto,,'t'));
vto=tranwrd(vto, "ctr", "center");
vto=tranwrd(vto, "medical", "med");
* vto=tranwrd(vto, "med", "medical");
* medical center = ?;
vto=tranwrd(vto, "texas", "tx");
vto=tranwrd(vto, "hospital", "hosp");
```

Validate your approximate link

```
data table1;
linkv=compress(lowcase(lname));

data table2;
linkv=compress(lowcase(lname));

data linked; * approximate link;
merge table1 table2 (rename=(lname=lname2));
by linkv;

proc print data=infn(obs=100);
where lname~=lname2;
```

Take Away

- When merging data
  - Use numeric codes whenever possible
  - Remember to use uniform formatting
  - Use string functions to standardize variables
  - Check if the key provides unique rows
    - 1-to-1 or 1-to-N mapping
- Pay attention to what rows link and what do not
- Consider how many rows should link
  - Example: 20% expected 18% achieved
- Validate by printing
  - Links made
  - Links not made

Lab 4

- Answer posted on website
- Look at how I compared using excel

```
proc transpose data=append2 out=data.tappend prefix=week;
      id week;

proc transpose data=append2 out=data.tappend prefix=week;
proc transpose data=append2 out=data.tappend;
      id week;
```

Debugging in practice

- Run through computer code on paper
  - Basically write variables
  - Track how it is changing

What you learned so far…

- Assignment 1
  - Setup work environment
  - Use the SAS software
  - SAS programming basics
    - data step & proc step
    - libname
    - Writing code & Reading logs
- Assignment 2
  - Understand variables (names, types, labels)
  - To write conditional logic codes
  - Subset columns (variables) from a table
  - Subset rows (observations) from a table
  - Recode, rename variables and calculate new variables
  - Label variables and values

What you learned so far…

- Assignment 3
  - use for loops (iterative loops)
  - use while loops (conditional loops)
  - SAS: use one dimensional arrays
- Assignment 4
  - **Append multiple tables (more rows)**
    - **stack tables on top of each other to increase the number of rows**
    - using `set`

What you learned so far…

- Assignment 4 continued
  - **Link up multiple tables using a shared key (more columns)**
    - **align the rows using the shared key, and link multiple tables to increase the number of variables in the tables**
    - using `merge`
    - Be sure to understand the different behavior given different situations (i.e. what happens to shared vars? What happens to not shared vars?)
      - What is a 1-to-1 link
      - What is a 1-to-N link
      - What is a N-to-N link (you will not be doing this, but need to understand what this is. This must be done with proc sql in SAS)
  - Combine multiple rows into one row
    - by group processing `proc summary`
  - Reshape table to flip rows & columns
    - using `proc transpose`
    - Also transpose (flip rows & columns) by groups or row

Table Operations: 1 table → 1 table (reshaping)

- Proc Transpose

1	2
a	d
b	e
c	f

→

1	a	b	c
2	d	e	f

- Proc Summary

A
B
C

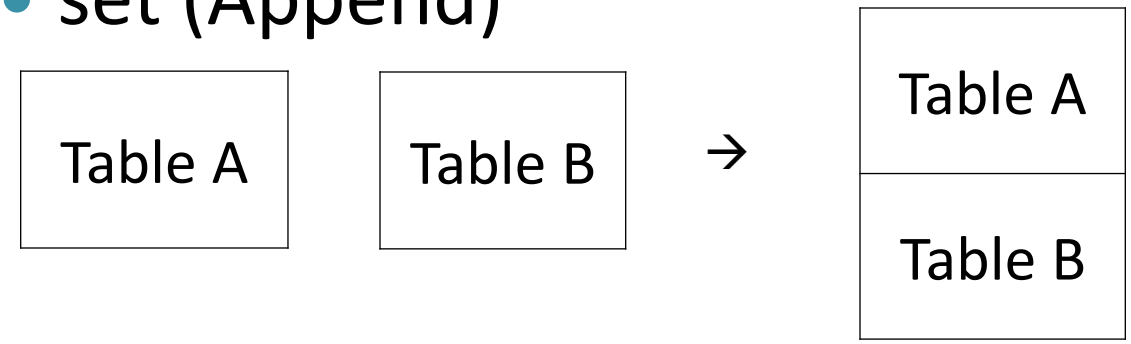
→

D

Where D=function(A,B,C)
Examples of function are
Sum(A,B,C) Mean(A,B,C) Max(A,B,C) Min(A,B,C)

Table Operations: multiple table → 1 table

- set (Append)



- merge (link)