Deterministic and Probabilistic Matching

Jared Edgerton

Note on data. This problem set uses **synthetic** (simulated) individual-level data. The goal is to practice record linkage and threshold selection—not to draw substantive inferences about real people.

Conceptual Questions

Please write three to ten sentence explanations for each of the following questions. **You are only required to answer ONE of the two questions below.**

1. Explain why deterministic (exact) matching can fail even when two datasets contain the same underlying individuals. In your answer, define the trade-off between **false matches** and **missed matches**, and explain how probabilistic matching (e.g., `fastLink`) attempts to manage that trade-off.
2. Matching errors are often **not random**. Give two reasons why record linkage error might vary across individuals or groups (e.g., name commonness, transliteration, data-entry error, moving/ZIP changes). Explain one implication for social science inference if linkage quality differs systematically across groups.

Applied Exercises

Use the code in the week's code tutorial and the lecture slides to answer the following questions.

3. **Generate (or load) the synthetic data + deterministic matching.** Run the provided script to generate and save `dataset_a.csv` and `dataset_b.csv`, then load them into R.
   - Perform deterministic (exact) matching on `firstname`, `lastname`, `birthyear`, and `zipcode` (e.g., `merge(..., by = ...)`).
   - Report the number of deterministic matches and the match rate (matches divided by `nrow(df_a)`).
   - In 3–6 sentences, explain why the deterministic match count looks the way it does in this simulation.
4. **Probabilistic matching with `fastLink` + threshold curve.** Using `fastLink`, match `df_a` and `df_b` on `firstname`, `lastname`, `birthyear`, and `zipcode`.
   - Use `fastLink(..., return.all = TRUE)`.
   - Use `getMatches(..., threshold.match = t)` for a grid of thresholds from 0 to 1 (e.g., increments of 0.01).
   - Create a plot of **number of matches** vs. **threshold**.
   - In 4–6 sentences, describe how and why the curve changes as the threshold increases.
5. **Match quality, choosing a threshold, and interpreting posteriors.** Using the probabilistic matches, evaluate match quality as the threshold changes and justify a final choice.

- Create a low-threshold "candidate match" set (e.g., `threshold.match = 0.000001`) and then group matches by posterior bins (e.g., 0.0–0.1, 0.1–0.2, . . . , 0.9–1.0).
- For each posterior bin, compute:
  (a) Levenshtein distance for first names (e.g., `stringdist(..., method = "lv")`),
  (b) Levenshtein distance for last names,
  (c) absolute difference in birth year.
- Make at least one plot that shows how these distances relate to posterior scores (e.g., boxplots of distance by posterior bin).
- Based on your diagnostics, choose a threshold you would use for this dataset and defend your choice (5–8 sentences).
- In your discussion, address:
  (a) how deterministic vs probabilistic matching differ in the number of matches found,
  (b) how changing the threshold affects both the number of matches and match quality,
  (c) at least two limitations or biases of each approach,
  (d) the relationship between string distance and the posterior/threshold measure.

6. **Challenge Question (Optional — if you finish early):** Experiment with modelling choices in `fastLink`.
   - Try at least **two** different matching-variable sets (e.g., drop `zipcode`; or match only on names + birthyear).
   - Try at least **two** different string distance methods for names (see `stringdist` methods such as Jaro-Winkler vs Levenshtein).
   - For each configuration, plot the threshold–matches curve and summarize how match quality changes (5–8 sentences).
   - Briefly discuss the implications of these modelling choices for real social-science linkage tasks (e.g., voter files, administrative records, platform data).