dedupe

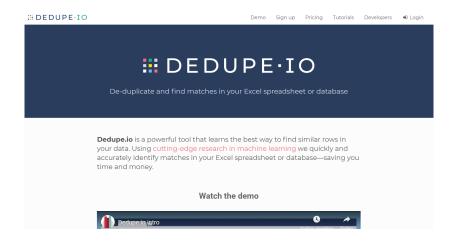
## A common problem

- ► Task: Do two names refer to the same person?
- Examples
  - A. Abel, Andrew Abel and Andrew B. Abel
  - Bill Christie and William G. Adams
  - Xiong Chen and George Chen
  - Darrell Duffie, Darel Duffie and Darrell Daffie

#### How I went about this

- 1. Look up each person manually
- 2. Sort by surname, then first name, use already collected information
- Use predefined name, change it only for certain instances after internet search
- 4. Use difflib library to suggest matches to new entries
- 5. Use difflib library to suggest matches to new entries whose last name starts with the same letter

## dedupe, a Python library to link records and deduplication



# Functioning of dedupe

- Probability is weighted distance of field-entries
- ▶ Field weights are *learned* by algorithm
- Solve entries manually that are most uncertain of being duplicates (why?), then relearn weights → Active Learning

# Functioning of dedupe

- Probability is weighted distance of field-entries
- ► Field weights are *learned* by algorithm
- Solve entries manually that are most uncertain of being duplicates (why?), then relearn weights → Active Learning
- ▶ Reduce number of pairs by grouping possible pairs after learning → blocking rules
- Cluster possible groups of pairs after estimating their matching probability
- lacktriangle Define matching threshold as F-score computed from ightarrow Precision and Recall

# Using dedupe as Programmer

- https://dedupe.io/developers/library/en/latest/ index.html
- ▶ pip install dedupe
- 1. Instantiate with list of field definitions (dict)
- 2. Feed with data organized as index-oriented nested dict
- 3. Train
- 4. Match uncertain rows manually
- 5. Learn again
- 6. Merge back to data