Coding task for Princeton Eviction Lab Data Engineer position

May 30, 2025

**Introduction**

People and companies often list multiple misspellings or versions of the same address. For example, at different places the same person may write “10 Nassau Street Princeton NJ 08540” as:

* “10 Nassau Princeton NJ 08540”
* “10 Nassua Street Princeton NJ 08540”
* “10 Nassau Street NJ”
* Etc.

Even after cleaning and parsing these addresses, it is often difficult to impute missing information or determine misspellings for idiosyncratic street names on a row-by-row basis. (e.g., It’s hard to know 10 Nassau Street is in Princeton, since there are many other Nassau Streets in the US, or that Nassua should actually be Nassau). One way to fix these errors is to use the other addresses that an individual is connected to to make corrections. In a vacuum, we cannot know that “Nassua” should be “Nassau”, but if there are 10 instances of “10 Nassau Princeton NJ 08540” and one instance of “10 Nassua Princeton NJ 08540” we can be fairly confident that Nassau is correct. The goal of this task is to write a script that uses the full range of addresses an individual is connected to to clean the information within them.

**Datasets**

Included in this folder are the following datasets. All of the datasets are from public tax assessment and business filing records and their use and dissemination has been approved by Princeton’s Institutional Review Board. They consist of a sample of 100,000 individuals and their contact addresses from the Boston metro area. The addresses may be from anywhere, though, since Boston companies and property owners may have contact addresses outside of the Boston area.

**fe.csv**: entities: each row constitutes a single person or company, and the dataset includes the following fields

* EID: entity ID, a unique identifier for each person or company
* fullName: the individual’s full name
* entityType\_basic: an indicator of whether they are a person, company, or government entity
* entityType\_legal: an indicator of their legal type (e.g., corporation, LLC)
* entityType\_subst: an indicator of their substantive characteristics (e.g., realty company, charitable organization, etc.)

**fa.csv**: addresses: each row constitutes a unique address, as defined by its “fulladdress\_c” field. Accordingly, “10 Nassau Street” is considered to be a distinct address from “10 Nassau Street Princeton” the components have already been cleaned and (although there are plenty of outstanding errors) do not need to be cleaned further except through the use of the other addresses, as described below)

* AID: address id, a unique identifier for each address
* fullAddress\_c: the full address (1:1 relationship with AID)
* num1\_c: the street number in the address
* num2\_c: the second street number, if one exists
* streetDirectionalPre\_c: a directional indicator that precedes the street name, e.g. North Main Street
* streetName\_c: the street name
* streetSuffix\_c: the street suffix
* streetDirectionalPost\_c: a directional indicator that follows the street name, e.g. Main Street North
* zip\_c: the zip code (an underscore is included to prevent it being stored without preceding zeroes, we can discuss whether this is a smart approach in subsequent interviews!)
* unit\_c: the apartment or unit number, if one exists
* city\_c: the city or town
* country\_c: the country
* state\_c: the state
* po\_box\_c: the PO box number, if the address is a PO box

**r\_fe\_fa**: relationships between entities and addresses

* EID\_1: entity ID
* AID\_2: address ID
* relationshipType: whether the address is a contact address (an address listed by a person or company in the tax assessment records, or by an officer in the business filings) or a principal address (an address listed by a company as its principal location in the business filing records)
* number: the number of times that address was listed (e.g., if a person appears in 10 years of property records, always listing the same contact address, this will be 10)

**Description of task**

Please write a script to use the full range of addresses that each individual is connected to to fill in and correct information within them. You do not need to do any string cleaning or parsing of the address components. For example, even though there are instances where the street name or suffix has misspellings that could be corrected on a row-by-row basis, just ignore these and focus on fixes that draw on information from other addresses. We don’t want you to spend your time doing basic cleaning. Specifically, implement the following rules:

* For an address missing a zip code, if an entity it is connected to has other addresses with the same number and street name, all of which have the same (or missing) zip code, fill in that zip code for all missing. If there are multiple zip codes associated with the number and street, do nothing.
* If an entity is connected to multiple addresses with the same street number and street names that are within a string distance of 0.10 of one another, correct them all to the street name that is most common.
* Observe the data and come up with two other, similar rules.

Provide code to evaluate the extent of changes and to sample and observe the changes for accuracy.

Take care to consider that multiple entities may be connected to the same address. Accordingly, there may be two entities connected with “10 Nassau Street”, where one entity’s other addresses suggest that the full address is “10 Nassau Street Princeton” while the other’s suggest that it is “10 Nassau Street Boston”. Accordingly, it may be necessary to split what was one address into two addresses and re-index the address data. (Whether this is a smart way to store address information is something we can certainly discuss in the future!)

The script will be evaluated for:

* Effectiveness (it must complete the task as described, and more effective custom rules will be judged more highly)
* Extendability (writing the code in a way that other similar rules could be applied within the same framework)
* Clarity (please include comments, but no need to go crazy)
* Speed (The full datasets contain several million entities, so using a for loop to go line by line is likely to be too slow)

We anticipate this should take no more than three or so hours, so please do not go crazy. And please feel free to contact me with any follow-up or clarification questions.