Technical Document: Data Cleaning and Generation Process
Albemarle and Charlottesville Eviction Data
Cases Filed From July 1, 2018 - June 30, 2021

Jacob Goldstein-Greenwood, Michael Salgueiro, and Michele Claibourn

10/1/2021

We begin with data scraped from Virginia Judiciary Online Case Information System for the General District Courts.

The 2018 Q3/Q4, 2020, and 2021 data were gathered with a scraper developed by civic programmer Ben Schoenfeld; the code for that tool is available at the following link: https://github.com/bschoenfeld/va-court-scraper. Anonymized data are posted periodically to virginiacourtdata.org, and de-anonymized data—which we used to develop the eviction database—can be requested from Schoenfeld.

The 2021 Q1/Q2 data were gathered with a scraper developed by the University of Viginia (UVa) Legal Data Lab (LDL).

From the combined data, we subset unlawful detainer (eviction) cases filed in the Albemarle and Charlottesville General District Courts that have filing dates from 2018-07-01 to 2021-06-30.

We apply an aggregation and cleaning process to the data to improve the accuracy and reliability of later analyses. Each element of the process is documented and commented in the code files delivered to KAG/TJPDC, and we describe the steps below. Further, as part of the cleaning process, we generate a number of new fields reflecting key details about the cases (e.g., presence of defense attorney; total costs awarded; etc.).

Aggregation and cleaning

1. We first aggregate data up to the case level such that 1 row = 1 case
2. We clean and standardize the names of the primary (i.e., first-listed) plaintiff and defendant in each case: These are the `defendant_1_standardized` and `plaintiff_1_standardized` columns
    - Differences in court clerk data-entry styles, as well as typographical errors, threaten the baseline accuracy of analyses that involve grouping on or aggregating by plaintiffs/defendants (for example, one case may list a plaintiff as "SMITH-JONES PROPERTIES, L.L.C," and another may list that same plaintiff as "SMITH JONES PROPERTIES LLC"
    - To the best of our ability, we want to ensure that those cases can be associated with the same entity, and we therefore apply the following standardization process to plaintiff and defendant names:
        - Remove "T/A" and "D/B/A" tags (e.g., "JANE DOE LEASING T/A DOE LEASING" –> "JANE DOE LEASING")
        - Correct comma misplacements (e.g., "SMITH PROPERTIES , INC." –> "SMITH PROPERTIES, LLC")
        - Remove dashes, slashes, and periods (e.g., "SUPER-HOME APARTMENTS, L.L.C" –> "SUPER HOME APARTMENTS, LLC")

  - Remove trailing commas at the ends of names (e.g., "DOWNTOWN APARTMENTS," –> "DOWNTOWN APARTMENTS")
  - Remove all errant double, triple, etc. spaces (e.g., "MIDTOWN LEASING" –> "MIDTOWN LEASING")
  - Apply a standardization routine that converts common alternative spellings of certain words into one form (e.g., "MGMT/MGT/MTG" –> "MANAGEMENT")
  - Apply a set of typo corrections specific to these data (e.g., "OWNER" is misspelled as "OWENER")
  - Remove commas and semicolons that come before a business entity identifier (e.g., "WILLIAMS, LLC" –> "WILLIAMS LLC")
- Note that we also store original plaintiff/defendant names, exactly as they are read from court websites, in the database: These are the `defendant_1_unmodified` and `plaintiff_1_unmodified` fields
- We also generate fields containing *further-simplified* versions of defendant and plaintiff names by removing middle initials, JR/SR, I/II/III/IV, odd punctuation marks, etc.: These are the `defendant_1_simplified` and `plaintiff_1_simplified` columns

3. We then extract and clean defendant and plaintiff ZIP Codes from case records: These are the `defendant_1_zip` and `plaintiff_1_zip` columns
   - We treat as the primary ZIP Code for a given case the ZIP Code associated with the *first-listed (primary) defendant* (`defendant_1_zip`)
   - Note that publicly available data do not include the exact address of the property under dispute; instead, the data include an address for each defendant and plaintiff in a given case, granular to the level of *ZIP Code + locality* (e.g., "22902 CHARLOTTESVILLE, VA")
   - We convert non-VA and invalid ZIPs to NA so that they do not disrupt by-ZIP tabulations
     - We treat ZIPs within the following sets to be valid VA ZIPs: [20100, 20199] and [22000, 24699]
4. We then identify and remove true duplicates (e.g., errant double-entries by court clerks and errant double-downloads by data scrapers)
   - When we find multiple cases with the same filing date, judgment, costs, attorney fees, principal value, other awarded amount, primary plaintiff, primary defendant, and defendant ZIP, we remove all but one of the cases

---

Generation of new data fields

1. We identify and flag *serial* cases
   - Associated column in data: `serial_filings_by_plaintiff_against_defendant` (values: `TRUE` or `FALSE`)
   - We define serial cases as: Multiple cases in a 12-month period in which a plaintiff (`plaintiff_1`) filed against a defendant (`defendant_1`) within ZIP Code (`defendant_1_zip`)
   - When we identify a chain of serial cases associated with a given plaintiff/defendant/defendant ZIP combination, we flag that combination's cases with `TRUE` values in the `serial_filings...` column referenced above
2. We generate a column indicating whether a defense attorney is present in the case
   - Associated column in data: `defense_attorney_present` (values: `TRUE` or `FALSE`)
3. We generate a column indicating whether each plaintiff is likely a management company/business entity/quasi-governmental organization as opposed to an individual person (or a set of individuals; e.g., a couple)

- Associated column in data: `mgmt_company_plaintiff` (values: `TRUE` or `FALSE`)
- We identify likely management company/business entity/quasi-governmental organization plaintiffs using a regular-expression pattern developed in-house (see the `tjpdc-mgmt-company-regex.R` file)

4. We add a column indicating whether a judgment has been issued in the case by the end of the study period of interest (2021-06-30)
    - Associated column in data: `judgment_issued` (values: `TRUE` or `FALSE`)
5. We generate a column indicating whether the case only has one hearing associated with it
    - Associated column in data: `single_hearing` (values: `TRUE` or `FALSE`)
6. We determine the total dollar amount of judgments
    - There are five relevant columns in the scraped data: `Costs`, `AttorneyFees`, `PrincipalAmount`, `OtherAmount`, and `OtherAwarded`
        - The first four contain numeric values (e.g., 403.50)
        - The fifth (`OtherAwarded`) contains character strings in which additional awards/award details can be communicated (e.g., "$201.00 FOR DAMAGES")
            * We extract numeric values from `OtherAwarded` and place them in a column called `OtherAwardedVal`
    - We then sum the `Costs`, `AttorneyFees`, `PrincipalAmount`, `OtherAmount`, and `OtherAwardedVal` columns to generate a `total_judgment_amount` column (numeric)
    - Note that the `OtherAwarded` column sometimes doubles the information in `OtherAmount` (e.g., `OtherAmount = 89.00`; `OtherAwarded = '$89.00 FOR DAMAGES'`)
        - When we detect identical values in those two columns, we exclude `OtherAwardedVal` when when calculating `total_judgment_amount`
7. We generate a column containing the writ of eviction filing dates, when available, for cases
    - Associated column in data: `writ_issued_date` (yyyy-mm-dd)
8. We generate a column indicating whether each defendant is likely to be a residential defendant ("JOHN SMITH") or a non-residential entity (business, government body, etc.; e.g., "JONH SMITH INC."; "STATE OF VIRGINIA"; etc.)
    - Associated column in data: `non_residential_defendant` (values: `TRUE` or `FALSE`)
    - We identify likely non-residential defendants using a regular-expression pattern developed in-house (see the `tjpdc-non-residential-regex.R` file)

We write out two CSVs containing cleaned data:

1. `cases.csv`, which contains all cases
2. `cases_residential_only.csv`, which contains only those cases with `FALSE` in the `non_residential_defendant` field

---

Our full code for cleaning and summarizing data is available in the file `tjpdc-clean.R`.