Preparing Contra Costa County Charge Data for the ACLU of Northern CA
Chris Kaiser-Nyman

This project contains the following documents:

1. This **readme.pdf**
2. **“Contra Costa Court Data [Depersonalized].xlsx”** - (this file is not required to reproduce my efforts). If you have downloaded this project through Github, this file is not in the main Github repository file location, but in a release found here. If you have downloaded this project through google drive, the file should be with the rest of the files for this project. This is the original dataset provided to the Contra Costa County Public Defender’s office by the Contra Costa County Department of Information Technology, that Umesh and Dhruv manually cleaned to produce the file below
3. **"Contra Costa Court Data (Depersonalized) - Manually Cleaned.csv"** - If you have downloaded this project through Github, this file is not in the main Github repository file location, but in a release found here. If you have downloaded this project through google drive, the file should be with the rest of the files for this project. This file was created by Dhruv Madaan and Umesh Thillaivasan as described by them below:
    a. We originally received a depersonalized 305 MB Excel file (Contra Costa Court Data [Depersonalized].xlsx) from Jeff Landau which contained 3 Header rows, 276,232 records, and 296 columns (or features). The file received required a lot of data cleaning, and the first thing we noticed was that during the original query exportation, several columns had shifted by 1 or more columns to the right. We sorted the data by column entries where we had recognized problems, and manually shifted records by the appropriate number of columns. The Zip Code column was exported incorrectly where some values were missing leading characters or converted to massive scientific notation. For example, 94565 was often seen as 4.56E9 or 45650. The City column also had numerous problems which needed to be resolved. Since we had so many unique types of city entries, we could not use a fuzzy string comparison function because threshold matching differed from city match to city match. Instead, we manually combed over the City column and standardized the entries to address spelling and fuzzy string matches. This would let us have at least one reliable geolocation column to leverage. After shifting the columns and cleaning up the City column, we then were able to export the Excel file as a comma-separated file for further data wrangling. After making sure that the data was at least in the correct spreadsheet format, we saved the data as a csv (Contra Costa Court Data - Manually Cleaned.csv) file.
4. **initial_data_cleaning.ipynb** - a Python notebook initially created by Dhruv Madaan and Umesh Thillaivasan, two UC Berkeley graduate students for the engineering class IEOR 242. The notebook was edited slightly by me, Chris Kaiser-Nyman, with help from

Anderson Lam This Python notebook uses "Contra Costa Court Data (Depersonalized) - Manually Cleaned.csv" to create a file for use with the subsequent R Markdown file.

5. **ccc_charge_data_final.Rmd** - an R Markdown file created by me (Chris Kaiser-Nyman) used to continue cleaning the file created by the Python notebook, and add data
6. **chargecodes.csv** - a file imported from here on April 5, 2019 that allowed me to add descriptions and BCS codes (severity of the charge) for each charge
    a. .txt link
    b. .csv link
7. **ccc_charge_data_narrative.Rmd** - a file similar to the other R Markdown, but that shows more of the iterative process by which the code was developed and that also provides some steps and initial code that can be used for further analysis. This file should be run in a separate workspace from the other R markdown file, as it uses some of the same names for objects but does different things with them.
8. **python_output.csv** - the file that would result from running the python notebook. The actual output file from the python notebook will be titled “aclu_Nov_2014.csv”, but these two files should be identical. I am only including the python_output.csv file for those who do not have access to Python.
9. **r_markdown_output.csv** - the file that would result from running the R Markdown file. The actual output file from the python notebook will be titled "ccc_final_data.csv", but these two files should be identical. I am only including the r_markdown_output.csv file for those who do not have access to R.

Potential issues I have identified:

It is vital that someone review the code written in both the Python notebook and R Markdown file to ensure there are no errors. I do not know what the review process was for the Python notebook, but the R Markdown was written entirely by me with no code review. While all code runs, it is important that someone with knowledge of both the datasets and R review it to ensure that it is doing what it was written to do.

Initial data cleaning:

- The initial manual cleaning by Dhruv and Umesh is not reproducible, and therefore cannot be verified. I believe that the most likely source of error, if there is any, is that they were not able to successfully shift all rows and columns back to their correct places. Given that they had to manually inspect 276,232 records, each with 296 columns, it seems like this would be an easy source of error

Problem with the charge code key:

- When creating the “key” column in the “chargecodes” data frame that is the basis for merging descriptions of the charges and BCS codes, I had to paste some of the “offense” listings together, through the following code:

```
uniquechargecodes <- chargecodes %>%
        group_by(key) %>%
         summarise(unique_description = paste(offense, collapse = "----,---- "))
```

This would occur in any case where the “penal code” “offense number” and “m_f” columns are identical in the “chargecodes” dataframe. This may be a problem, or it might not.

Not all charge codes / BCS codes merge:

- When trying to link the Charge Codes of the original data to the descriptions and BCS codes of the chargecodes key provided by the CA Department of Justice here (https://oag.ca.gov/law/code-tables), I used Charge Code 1 as a benchmark for percent of charge codes matched from the original dataset to the key data. I used Charge Code 1 as the benchmark, because only 8 out of over 10,000 cases are NA for Charge Code 1, whereas Charge code 2 has 3,205, and each charge code beyond that increases the number of cases without an additional charge code. 15% of charges (for Charge Code 1) still do not have a match, so work should be done to improve that. A discussion of one possible route to increasing the match rate is presented in the “narrative” R Markdown file of this project.

NAs:

- NAs may be a problem. The original dataset contains blanks in some places where appropriate, and **possibly** in places there shouldn’t be. One example of somewhere that blanks would be appropriate is in the Charge Code 2 column. Any given person may only have one charge, so a blank or NA in Charge Code 2 would make sense. However, both the Python and R code may introduce NAs to columns where they would mean something different from the NAs or blanks that were there originally. For example, the original Charge Code 1 column contains 8 NAs. This doesn’t seem to make much sense because why would that entry exist if there are no charges associated with it. Furthermore, when creating a new column with the BCS codes for each charge, any time the code cannot find a matching Key for the Charge Code, it introduces an NA. These “no match” NAs are indistinguishable from the NAs that result from no charge existing for that case. This makes more sense in the case of Charge Code 2, where many cases legitimately have not second charge, but in the BCS_Charge_2 column (created in the R markdown file), NAs introduced because of not being able to find a matching key in the chargecodes codebook are indistinguishable from NAs as a result of there being no charge. This should be a relatively simple fix that would involve changing the code to output “No Charge” if there is an NA in the original Charge Code 2 column, and an NA if there was a charge, but the code was not able to match the key from the chargecodes codebook.

.

Potential for Analysis:

The output of the R markdown file ("ccc_final_data.csv" if you reproduce the results through the R Markdown, or “r_markdown_output.csv” if you just use the file provided) can be used for descriptive statistics in Excel or other software. One could determine the racial composition of everyone charged during any given time period (from November 2014 through February 2016), using the “Race” column, or gender through the “Gender” column. It is unclear to me whether the “Zip Code” and “City” columns represents the zip code & city the individual was arrested in or the zip code & city the individual resides in, but Jeff Landau of the Contra Costa County Public Defender’s Office should be able to help determine that, and once that is determined, the most common zip codes and cities could be identified using those columns. The additional columns “common zips” and “common cities” allow for analysis of just the most common areas, with all others grouped into “other”. Race, gender, city, and zip could also be used to provide descriptive statistics of the number of charges, enhancement of those charges a given case has, the jail and probation time for each charge, and the number of days between the incident (“Incident Date” column) and the date the charges were filed in court (“Court File Date”). As with all of these descriptive statistics, without any controls, the statistics are truly descriptive: for example, any differences between races in the number of charges against the individual could be due to the type of charges, the severity of those charges, or the agency involved (“Incident Agency”).

The BCS Code columns could help classify the severity of crimes charged, but with only a 70% match rate for the BCS code for Charge 1, until it has a much higher match rate, using it even for descriptive statistics could be misleading if there are demographic differences in the kinds of charges that merged vs those that did not merge with the codebook.

Once close to 100% of BCS codes are able to be merged into the main dataset, I would expect that a number of interesting regressions could be run. Determining whether there are racial differences in jail time served, number of charges faced, severity of charges, and types of charges, would all be interesting to examine, using as many controls as the regression can support.

Next Steps:

Code review:

- It is vital that someone reviews the code for both the Python notebook and the R markdown files. While all of the code runs, it has not been reviewed to ensure it does what it is expected to do.

Improving codebook merge:

- BCS codes are only merging at 70%, because they do not have the manually-entered codes that the text-description codebook has (lines 107-116 in

“ccc_charge_data_final.rmd” or 152-164 in “ccc_charge_data_narrative.rmd”). Adding this would be a very simple fix to increase merging to 85%

- Improving the match rate to as close as possible to 100% could be done through code, as described in lines 204-254 in “ccc_charge_data_narrative.rmd”, or could be done manually as described above
- Charge Enhancements should also likely be be merged with text descriptions and BCS codes as with the Charge Codes. However, the Charge Enhancements do not have the same “M” & ““F” suffixes as the Charge Codes. Assuming that these codes are the same as the "Charge Codes" codes, the only hiccup here is that "M" & "F" have different BCS codes, so getting some advice as to how to classify the severity of these enhancements will be important. Merging of the BCS codes & written descriptions for all 55 "Charge Enhance" columns should be done to allow controlling for the severity of enhancements.

Determining the meaning of different columns

- The meaning behind much of this data is unclear to me. For example, do the Zip Code and City data refer to wher the arrest took place, or where the defendent resides? What does the “Count” column mean (there are cases where the column “Count (1)” (or “Count_1_”) is “6” but there are no enhancements on the first charge, and no second or other charges. I also do not know whether the Jail and Probation time columns refer to time sentenced, time served, or something else.