MethodologyPart2

May 7, 2022

[1]:
```
%%html
<style>
table {float:left}
</style>
```

```
<IPython.core.display.HTML object>
```

[2]:
```
import sys

# append the directory of law module to sys.path list
sys.path.append('../modules/')
```

[3]:
```
import altair as alt
import arrest
import charge
import law
import pandas as pd
```

I'm going to import a lot of data here to mitigate clutter in the rest of the notebook, but I'll annotate them here for reference. ***

`oag_data` is a CSV published by the California Office of the Attorney General under the name "Offense Codes (with LEI codes)." It has no headers, but these are available in an XML file (`oag_xml`) published on the same page.

[4]:
```
oag_xml = pd.read_xml('../CA/01_inputs/chsoff_20220415.xml')
```

[5]:
```
oag_data = pd.read_csv('../CA/01_inputs/chsoff_20220415.csv', header=None,
                       names=oag_xml.columns,
                       usecols=['StatutoryNumericCodes',
                                'TypeOfStatCode',
                                'StatuteLiteral',
                                'TypeOfCharge',
                                'EnactDate',
                                'RepealAmendDate'],
                       parse_dates=['EnactDate',
                                    'RepealAmendDate'],
                       dtype=str)
```

`processsed_oag_data` is the output of normalization I performed on `oag_data` in `../CA/02_clean/01_statute.ipynb`. I elaborate on this below.

[6]:
```
processed_oag_data = pd.read_json(
    '../CA/01_inputs/processed/c01_chsoff.json', dtype=str)
```

`sacramento_arrests` is Sacramento Police Department arrest data from which I'm importing a subset of fields to illustrate arrest charge decisions I made.

[7]:
```
sacramento_arrests = pd.read_csv(
    '../CA/Sacramento/04_outputs/c00_arrest_data.csv',
    usecols=['ARREST_NUM', 'ARREST_DATE', 'ARREST_TIME', 'CLASS',␣
 ↪'OFFENSE_D','FELONY_MISDEMEANOR'],
    dtype=str
)
```

`san_diego_arrests` is San Diego Police Department arrest data from which I'm importing a subset of fields to illustrate arrest charge decisions I made, and to compare with `sacramento_arrests`.

[8]:
```
san_diego_arrests = pd.read_csv(
    '../CA/San Diego/04_outputs/c00_charges.csv',
    usecols=[
        'activityNumber',
        'activityDate',
        'violationType',
        'violationSection',
        'chargeDescription',
    ],
    dtype=str,
)
```

1 Arrest Charges

1.1 Decision: Normalizing charge data

The California Office of the Attorney General publishes data on the range of offense levels associated with each particular charge code. I used the Offense Codes (with LEI codes) CSV.

The process by which I selected a subset of it is entirely captured in `../CA/02_clean/01_statute.ipynb`. In this notebook I'll show you the cooking show version.

From the California legislature website (annotated):

1.1.1 647(F)

PENAL CODE - PEN
PART 1. OF CRIMES AND PUNISHMENTS [25 - 680.4] *(Part 1 enacted 1872.)*

TITLE 15. MISCELLANEOUS CRIMES [626 - 653.75] *(Title 15 enacted 1872.)*

CHAPTER 2. Of Other and Miscellaneous Offenses [639 - 653.2] *(Chapter 2 enacted 1872.)*

647. Except as provided in paragraph (5) of subdivision (b) and subdivision (k), every person who commits any of the following acts is guilty of disorderly conduct, a misdemeanor:

[...]

(f) Who is found in any public place under the influence of intoxicating liquor, any drug, controlled substance, toluene, or any combination of any intoxicating liquor, drug, controlled substance, or toluene, in a condition that they are unable to exercise care for their own safety or the safety of others, or by reason of being under the influence of intoxicating liquor, any drug, controlled substance, toluene, or any combination of any intoxicating liquor, drug, or toluene, interferes with or obstructs or prevents the free use of any street, sidewalk, or other public way.

***

Parsed:

_code_type	_section	_subdivision
PC (PEN)	647	(F)

CHSOFF (processed_oag_data)

[10]:
```
processed_oag_data[processed_oag_data['_original_code'] == '647(F)'][
    [
        '_code_type',
        '_original_code',
        '_offense_level',
        '_charge_reconstructed',
        '_charge_description',
    ]
]
```

[10]:
```
    _code_type _original_code _offense_level _charge_reconstructed  \
103         PC         647(F)              M              PC647(F)

                                  _charge_description
103  DISRD CONDUCT:DRUG W/ALC, DISORD CONDUCT:ALCOH…
```

I'm using it for two reasons:

1. Charge data can be idiosyncratic, both across and within agencies.

San Diego

[11]:
```
san_diego_arrests[san_diego_arrests['violationSection'].str.contains('647(F)',␣
↪regex=False)].groupby(
    ['violationType', 'violationSection', 'chargeDescription']
)['activityNumber'].nunique().sort_values().reset_index()
```

[11]:
```
  violationType violationSection  \
0            PC         647(F)TO
1            PC         647(F)DG
2            PC         647(F)DK
3            PC         647(F)DK

                                  chargeDescription  activityNumber
0                          DISORDERLY CONDUCT: TOLUENE               7
1    DISORDERLY CONDUCT: DRUGS OR DRUGS WITH ALCOHOL             561
2  DRUNK IN PUBLIC: ALCOHOL, DRUGS, COMBO OR TOLU…            1394
3                          DISORDERLY CONDUCT: ALCOHOL            9836
```

Sacramento

[12]:
```
sacramento_arrests[sacramento_arrests['OFFENSE_D'].str.contains(
    '647(F)', regex=False)].groupby(['OFFENSE_D'])['ARREST_NUM'].nunique().
↪reset_index()
```

[12]:
```
                        OFFENSE_D  ARREST_NUM
0  647(F) DIP/REPEAT OFF 647D(B)         144
1     647(F) PC DIP - DISORDERLY       11699
```

2. It allows me to validate data entered.

PC 647(F) is listed exclusively as a misdemeanor offense, and I can validate the recorded offense level for the arrest by comparing it:

[13]:
```
sacramento_arrests[
    sacramento_arrests['OFFENSE_D'].str.contains('647(F)', regex=False)
].groupby(['OFFENSE_D', 'FELONY_MISDEMEANOR'])['ARREST_NUM'].nunique().
↪reset_index()
```

[13]:
```
                        OFFENSE_D FELONY_MISDEMEANOR  ARREST_NUM
0  647(F) DIP/REPEAT OFF 647D(B)                  M         144
1     647(F) PC DIP - DISORDERLY                  M       11699
```

1.2 Decision: Offense level

For every jurisdiction, my request followed this template regarding specific fields.

I request the following information for each arrest:

- Observation (rows): Arrest charge
- Time frame: 1/1/2009 to date of export
- Fields:
  - Any unique identifier for the arrest event, e.g. incident and/or case number
  - Any field indicating whether the arrest was made in response to a call for service
- Arrestee:
  - Name
  - Race
  - Ethnicity
  - Gender
  - Date of birth
  - Address recorded at time of arrest
- Offense and charge description
- Charge level (misdemeanor/felony class)
- Date and time of arrest
- Location of arrest (address and/or geographic coordinates)
- Police division
- Police beat

Four out of six cities provided offense/charge level data.

City	Offense Level Data
Los Angeles	Provided
Oakland	Provided
Sacramento	Provided
San Diego	Partially imputed
Portland	Provided
Seattle	Absent

1.2.1 Approach

Seattle I'm omitting Seattle frome felony analyses.

San Diego The chart goes into (perhaps too much) detail, but because these are all internal, I thought it best to risk redundancy.

I categorized a San Diego arrest as including a felony charge if, among the charges, at least one was only listed as having a felony offense level (e.g. homicide).

"Potential" felony charges I have also analyzed what the composition of arrests within housing status looks like if an arrest contains either a mandatory felony or a potential felony (San Diego**).** My current approach to the felony chart is including San Diego with an annotatation explaining these two variants.**

Chart TK

1.3 Decision: "Violent" charges

I don't have a programmatic description of my decisions here except that I:

- included codes under "person crimes" in each state.
- included municipal codes that entailed physical contact.

Here is a spreadsheet detailing charge codes categorized as violent.