Live Session Module 8

H. Diana McSpadden (hdm5s)

All court filings in district and circuit courts in Virginia are public, so any time an individual appears in court, the case is filed under that individual’s publicly accessible criminal record. That happens regardless of the outcome of the case: even dismissed charges appear on a person’s record. These records, especially convictions, are compiled by private companies that perform criminal background checks for rental, job, and loan applications. The effect is the creation of a permanent underclass of citizens in Virginia: people who have already served all legal punishments who continue to be punished by the lack of access to housing, employment, and financing.

Record sealing is the process by which a person’s criminal record is removed from the public record. It doesn’t destroy the records (that would be ”expungement”) as the records persist in a private database for use by state legal and law enforcement agencies, but it does prevent these records from appearing on non-governmental criminal record background checks. Until 2021, the only records that could be sealed for individuals in Virginia’s legal system were charges that were outright dismissed or applied to the wrong individual due to clerical errors or identity theft. Even then, to seal a record, an individual was required to use a lengthy and difficult petition process to do so. As such, very few records in Virginia have been sealed under the previous law.

In 2021 the Virginia state legislature passed a new lawLinks to an external site. allowing for automatic sealing of certain criminal records. The Legal Aid Justice CenterLinks to an external site. has taken on the work of lobbying state government to remove some of the restrictions included in the law for the purpose of simplifying the law, expanding the number of people who can benefit from criminal record sealing, and reduce the racial disparity among individuals who qualify for record sealing. Code for CharlottesvilleLinks to an external site. is assisting this effort by providing data analysis to (1) encode whether any given case can be automatically sealed, sealed by petition, or is ineligible for record sealing, (2) report frequencies of these outcomes statewide and within particular localities, and (3) consider how these frequencies would change given one or more hypothetical changes that can be made to the record sealing law in the future.

The data originate from Virginia’s Online Case Information System which allows anyone to look up any record by name, location, date, and other fields. The public data were compiled and made available for bulk downloadLinks to an external site. for the first time by Code for Hampton Roads volunteer Ben Schoenfeld. The files contain records of all cases that have appeared in the Virginia district and circuit courts since 2009 (with some courts providing data back to 2000). The data contain about 9 million cases from about 3 million different people. The following data file is a random sample of 100,000 people (anonymized) from this dataset with their whole records (about 350,000 rows total):

In [ ]:
```
import pandas as pd
import numpy as np
```

In [ ]:
```
# try to load the large dataset
df_100k = pd.read_csv('data100k.csv')
```

In [ ]:
```
df_100k.head()
```

Out[ ]:

	person_id	HearingDate	CodeSection	codesection	ChargeType	chargetype	Class	D
0	102090000000110	2019-02-28	A.46.2-862	covered elsewhere	Misdemeanor	Misdemeanor	1	
1	343221000000125	2009-12-07	B.46.2-301	covered elsewhere	Misdemeanor	Misdemeanor	1	
2	343221000000125	2011-01-20	A.46.2-707	covered elsewhere	Misdemeanor	Misdemeanor	3	
3	343221000000125	2011-07-01	B.46.2-301	covered elsewhere	Misdemeanor	Misdemeanor	1	
4	343221000000125	2012-10-15	B.46.2-301	covered elsewhere	Misdemeanor	Misdemeanor	1	

5 rows × 28 columns

Goals for this live coding session:

1. We will decide in class on some interesting questions we can answer with this dataset, such as

- What code sections are most frequent?
- Which ones most often lead to convictions?
- Which ones have the most severe racial disparities?
- In what localities (fips) are these disparities most severe?

2. But we will use pandas as needed to clean the data in specific, necessary ways prior to these analyses, including:

- Loading the data
- Recoding and collapsing categories
- Dropping extraneous columns
- Filtering rows to specific code sections or localities
- Aggregating to race, locality, and/or code section level data

In [ ]:
```
df_100k.head(1).T
```

Out[ ]:

	0
person_id	102090000000110
HearingDate	2019-02-28
CodeSection	A.46.2-862
codesection	covered elsewhere
ChargeType	Misdemeanor
chargetype	Misdemeanor
Class	1
DispositionCode	Guilty
disposition	Conviction
Plea	NaN
Race	Black(Non-Hispanic)
Sex	Male
fips	25
convictions	True
arrests	False
felony10	False
sevenyear	False
tenyear	False
within7	True
within10	True
class1_2	False
class3_4	False
expungable	Automatic (pending)
old_expungable	False
expungable_no_lifetimelimit	Automatic (pending)
reason	Conviction of misdemeanor charges listed in 19...
sameday	False
lifetime	False

What violations of the law have the greatest racial disparity?

- Black(Non-Hispanic)

In [ ]:

```
df_100k['Race'].value_counts()
```

Out[ ]:
```
White Caucasian(Non-Hispanic)                   114421
Black(Non-Hispanic)                              80173
White Caucasian (Non-Hispanic)                   41679
Black (Non-Hispanic)                             33254
Hispanic                                          9319
White                                             3527
Other(Includes Not Applicable.. Unknown)          3452
Asian Or Pacific Islander                         2787
Black                                             2200
MISSING                                           1022
Unknown (Includes Not Applicable.. Unknown)        785
Other (Includes Not Applicable.. Unknown)          615
American Indian                                    302
Unknown                                             54
Asian or Pacific Islander                            7
American Indian Or Alaskan Native                    1
Name: Race, dtype: int64
```

In [ ]:
```
df_100k['Race'].unique()
```

Out[ ]:
```
array(['Black(Non-Hispanic)', 'Hispanic', 'White Caucasian(Non-Hispanic)',
       'MISSING', 'Asian Or Pacific Islander', 'Black (Non-Hispanic)',
       'White Caucasian (Non-Hispanic)',
       'Other(Includes Not Applicable.. Unknown)',
       'Other (Includes Not Applicable.. Unknown)', 'Black', 'White',
       'Unknown (Includes Not Applicable.. Unknown)', 'American Indian',
       'Unknown', 'Asian or Pacific Islander',
       'American Indian Or Alaskan Native'], dtype=object)
```

In [ ]:
```
# Black (Non-Hispanic) & Black(Non-Hispanic) should be the same class
# Let's fix that
# make courts a copy of the df_100k
courts = df_100k.copy()

race_map = {'Black(Non-Hispanic)': 'Black',
            'Hispanic': 'Hispanic',
            'White Caucasian(Non-Hispanic)': 'White',
            'MISSING': 'Other',
            'Asian Or Pacific Islander': 'Asian or Pacific Islander',
            'Black (Non-Hispanic)': 'Black',
            'White Caucasian (Non-Hispanic)': 'White',
            'Other(Includes Not Applicable.. Unknown)': 'Other',
            'Other (Includes Not Applicable.. Unknown)': 'Other',
            'Black': 'Black',
            'White': 'White',
            'Unknown (Includes Not Applicable.. Unknown)': 'Other',
            'American Indian': 'American Indian or Alaskan Native',
            'Unknown': 'Other',
            'Asian or Pacific Islander': 'Asian or Pacific Islander',
            'American Indian Or Alaskan Native': 'American Indian or Alaskan Native
}

courts['Race'] = courts['Race'].replace(race_map)
```

In [ ]:
```
courts['Race'].value_counts()
```

Out[ ]:
```
White                               159627
Black                               115627
Hispanic                              9319
Other                                 5928
Asian or Pacific Islander             2794
American Indian or Alaskan Native      303
Name: Race, dtype: int64
```

What we need is to get a count of CodeSection and Race

In [ ]:
```
race_codesection = courts.groupby(['CodeSection','Race']).size().reset_index()
race_codesection = race_codesection.rename(columns={0:'Count'})
```

In [ ]:
```
race_codesection
```

Out[ ]:

	CodeSection	Race	Count
0	(74-4) 26-123	Black	1
1	01-2007	White	1
2	1	Black	5
3	1	White	3
4	1-12	Black	62
...	...	...	...
6635	Z.18.2-91	White	166
6636	Z.18.2-91; 26	Black	1
6637	Z.18.2-92	Black	1
6638	Z.18.2-95	Black	2
6639	Z18.2-47	Black	1

6640 rows × 3 columns

In [ ]:
```
# do a reshape - one column for each of the race groups - we want it wider
race_codesection_pivot = race_codesection.pivot_table(index=['CodeSection'],columns
```

In [ ]:
```
race_codesection_pivot
```

Out[ ]:

	Count					
Race	American Indian or Alaskan Native	Asian or Pacific Islander	Black	Hispanic	Other	White
CodeSection						
(74-4) 26-123	0	0	1	0	0	0
01-2007	0	0	0	0	0	1
1	0	0	5	0	0	3
1-12	0	0	62	0	0	13
1-200	0	0	26	0	1	17
...	...	...	...	...	...	...
Z.18.2-91	0	3	131	2	0	166
Z.18.2-91; 26	0	0	1	0	0	0
Z.18.2-92	0	0	1	0	0	0
Z.18.2-95	0	0	2	0	0	0
Z18.2-47	0	0	1	0	0	0

4207 rows × 6 columns

In [ ]:

```
# want to get rid of the multi-index on the columns
race_codesection_pivot = race_codesection_pivot.droplevel(0,axis=1)
```

In [ ]:

```
race_codesection_pivot
```

Out[ ]:

Race	American Indian or Alaskan Native	Asian or Pacific Islander	Black	Hispanic	Other	White
CodeSection						
(74-4) 26-123	0	0	1	0	0	0
01-2007	0	0	0	0	0	1
1	0	0	5	0	0	3
1-12	0	0	62	0	0	13
1-200	0	0	26	0	1	17
...	...	...	...	...	...	...
Z.18.2-91	0	3	131	2	0	166
Z.18.2-91; 26	0	0	1	0	0	0
Z.18.2-92	0	0	1	0	0	0
Z.18.2-95	0	0	2	0	0	0
Z18.2-47	0	0	1	0	0	0

4207 rows × 6 columns

In [ ]:

```
race_codesection_pivot = race_codesection_pivot.assign(total = race_codesection_piv
                                                        + race_codesection_pivot['As
                                                        + race_codesection_pivot['Bl
                                                        + race_codesection_pivot['Hi
                                                        + race_codesection_pivot['Ot
                                                        + race_codesection_pivot['Wh
```

In [ ]:

```
race_codesection_pivot
```

Out[ ]:

Race	American Indian or Alaskan Native	Asian or Pacific Islander	Black	Hispanic	Other	White	total
CodeSection							
(74-4) 26-123	0	0	1	0	0	0	1
01-2007	0	0	0	0	0	1	1
1	0	0	5	0	0	3	8
1-12	0	0	62	0	0	13	75
1-200	0	0	26	0	1	17	44
...	...	...	...	...	...	...	...
Z.18.2-91	0	3	131	2	0	166	302
Z.18.2-91; 26	0	0	1	0	0	0	1
Z.18.2-92	0	0	1	0	0	0	1
Z.18.2-95	0	0	2	0	0	0	2
Z18.2-47	0	0	1	0	0	0	1

4207 rows × 7 columns

In [ ]:

```
race_codesection_pivot = race_codesection_pivot.assign(perc_black = race_codesectio
race_codesection_pivot = race_codesection_pivot.assign(perc_white = race_codesectio
race_codesection_pivot = race_codesection_pivot.assign(disparity = (race_codesectio
```

In [ ]:

```
race_codesection_pivot
```

Out[ ]:

Race	American Indian or Alaskan Native	Asian or Pacific Islander	Black	Hispanic	Other	White	total	perc_black	perc_white	di
CodeSection										
(74-4) 26-123	0	0	1	0	0	0	1	1.000000	0.000000	1.
01-2007	0	0	0	0	0	1	1	0.000000	1.000000	-1.
1	0	0	5	0	0	3	8	0.625000	0.375000	0.
1-12	0	0	62	0	0	13	75	0.826667	0.173333	0.
1-200	0	0	26	0	1	17	44	0.590909	0.386364	0.
...	...	...	...	...	...	...	...	...	...	
Z.18.2-91	0	3	131	2	0	166	302	0.433775	0.549669	-0.
Z.18.2-91; 26	0	0	1	0	0	0	1	1.000000	0.000000	1.
Z.18.2-92	0	0	1	0	0	0	1	1.000000	0.000000	1.
Z.18.2-95	0	0	2	0	0	0	2	1.000000	0.000000	1.
Z18.2-47	0	0	1	0	0	0	1	1.000000	0.000000	1.

4207 rows × 10 columns

we don't care if the total is low, we need statistical significance

In [ ]:

```
# filter the rows if the values are low
race_codesection_pivot_filtered = race_codesection_pivot.query("total >= 200")
race_codesection_pivot_filtered
```

Out[ ]:

Race	American Indian or Alaskan Native	Asian or Pacific Islander	Black	Hispanic	Other	White	total	perc_black	perc_white	di
CodeSection										
16.1-253.2	0	0	56	1	0	152	209	0.267943	0.727273	-0.
17-7	0	10	43	0	0	151	204	0.210784	0.740196	-0.
18.2-102	0	1	129	2	1	127	260	0.496154	0.488462	0.
18.2-103	1	26	1295	31	17	2164	3534	0.366440	0.612337	-0.
18.2-104	0	2	266	1	7	349	625	0.425600	0.558400	-0.
...	...	...	...	...	...	...	...	...	...	
C.46.2-894	2	4	91	2	6	216	321	0.283489	0.672897	-0.
C.46.2-896	0	5	99	3	4	273	384	0.257812	0.710938	-0.
MISSING	0	0	129	5	8	104	246	0.524390	0.422764	0.
NO DMV	0	4	175	16	6	202	403	0.434243	0.501241	-0.
Z.18.2-91	0	3	131	2	0	166	302	0.433775	0.549669	-0.

145 rows × 10 columns

when is % higher for blacks than whites and when it is reversed.

In [ ]:

```
race_codesection_pivot_filtered.sort_values(by='disparity',ascending=False)
```

Out[ ]:

Race	American Indian or Alaskan Native	Asian or Pacific Islander	Black	Hispanic	Other	White	total	perc_black	perc_white	di
CodeSection										
29-48	0	1	228	0	1	30	260	0.876923	0.115385	0.
18.2-53.1	0	5	1411	16	5	309	1746	0.808133	0.176976	0.
46.2-938	0	1	319	7	2	69	398	0.801508	0.173367	0.
18.2-58	0	6	1142	18	10	400	1576	0.724619	0.253807	0.
18.2-32	0	1	197	3	1	74	276	0.713768	0.268116	0.
...	...	...	...	...	...	...	...	...	...	
4.1-305	3	26	444	41	29	2245	2788	0.159254	0.805237	-0.
54.1-3466	0	4	100	2	2	709	817	0.122399	0.867809	-0.
18.2-374.1:1	0	0	60	1	25	583	669	0.089686	0.871450	-0.
18.2-258.1	0	1	83	2	13	767	866	0.095843	0.885681	-0.
29.1-735	0	2	13	4	1	194	214	0.060748	0.906542	-0.

145 rows × 10 columns

In [ ]: