1. Voter File + Philadelphia Court Records Merge

The Merging Process

To address this uncertainty, we use fastLink: in addition to accounting for the uncertainty process, which can be adjusted for in subsequent empirical analyses, fastLink is transparent, open-source, and makes use of a scalable algorithm that allows for the merging of large datasets. Merging the nationwide voter file to the Philadelphia court records involves first conducting the within-state merge followed by the across-state merge.

The Within-Pennsylvania Merge

The aim of the within-state merge is to link the records of individuals who remained in Pennsylvania from the time of their court record appearance to the time when the voter files were updated. These individuals include those who remained in the same residence and those who moved within the state and updated their address in the voter file. Since a voter file for each state contains millions of records, we further reduce the scale of each merge process by additionally blocking observations on gender. To perform each merge, we use the following linkage fields which are present in both datasets:

- First Name
- Last Name
- Middle Initial
- Age

To make a comparison between the values of each linkage field across datasets, we use three levels of agreement for the string valued variables (first name, last name) based on the Jaro-Winkler distance with 0.85 and 0.94 as the thresholds. We also use three levels of agreement for age based on the absolute distance between values, with 6 months and 1 year as the thresholds for separate agreements, partial agreements, and disagreements, respectively. For the remaining variables (i.e., middle initial), we utilize a binary comparison based on exact matching, indicating whether they have an identical value. Specifically, for each one of the gender blocks, we used the following code:

```
library("fastLink")
matches.out <- fastLink( dfA = subset.1, dfB = subset.2,
                         varnames = c("first_name", "last_name",
                                      "middle_initial", "age"
                                      ),
                         stringdist.match = c("first_name", "last_name"),
                         numeric.match = c("age"),
                         partial.match = c("first_name", "last_name",
                                           "age"),
                         cut.a = 0.94,
                         cut.p = 0.85,
                         cut.a.num = 0.5,
                         cut.p.num = 1
                       )
```

Listing 1: Within-state merge via fastLink

In the above code, `subset.1` (`subset.2`) represents the subset of a gender block for the Pennsylvania (voter file) data. The names of the three variables used in the within-state merge are specified in `varnames`, while `stringdist.match` and `numeric.match` contain the list of variables that will be compared using string and numeric distance measures, respectively. The `partial.match` argument contains the list of linkage fields for which we make a comparison using three discrete levels (agreement, partial agreement, and disagreement). Finally, the `cut.a`, `cut.p`, `cut.a.num`, and `cut.p.num` arguments specify the thresholds used for the string and numeric distance comparison. For more details on these options and extra features of fastLink, please see `https://cran.r-project.org/package=fastLink`.

The Across-State Merge

The main problem of the within-state merge is the possible failure to match individuals who were residents of other states. In an effort to locate these individuals, we merged the sample of court records from Philadelphia with the within-state matching probability less or equal to 0.85 with all other states' voter files.

To compare the values of each linkage field across two datasets, we used the binary agreement variable for the string valued variables (first name and last name) based on the Jaro-Winkler distance with 0.94 as the threshold. We also used the binary agreement variable for age based on the absolute distance between values, with 6 months as the threshold used to separate agreements

and disagreements. The code for the across-state merge is given below.

```
library("fastLink")
matches.out <- fastLink( dfA = not.found.within, dfB = voter.file,
                         varnames = c("first_name", "middle_initial", "last_name", "age"),
                         stringdist.match = c("first_name", "last_name"),
                         numeric.match = c("age"),
                         cut.a = 0.94,
                         cut.a.num = 0.5,
                         threshold.match = 0.99
                       )
```

Listing 2: Across-state merge via fastLink

In the above code, `not.found.within` represents the subset of court records that could not be successfully matched in the within-state match step. The dataset `voter.file` is the full voter file without any subsetting. The remaining options in fastLink can be described in a similar fashion as to those used for the the within-state merge, with the exception that string and numeric comparisons were made based on two agreement levels, either agree or disagree — in other words, we did not use partial matching.

2. IRS Zip code-level income Income + Philadelphia Court Records Merge

We augment our data with a proxy measure of defendants' resources: zip code average income from the 2008 release of the IRS' Statistics of income. We merge in this data on zip code and construct terciles, following the code below.

```
# Code for merging in zip code income

# load functions
directory <- "../functions/"
functions <- list.files(directory)
loadfunctions <- sapply(functions, FUN = function(x)source(paste0(directory, x)))

# load IRS data
IRS <- read.csv("IRSzipcode2008.csv")
IRS <- subset(IRS, state == "PA")
# compute average AGI within zipcode
IRS %>% group_by(zipcode) %>% summarise(returns = sum(n1), AGI = sum(a00100)) -> resourcesData
resourcesData$avgAGI <- resourcesData$AGI/resourcesData$returns
resourcesData$zipcode <- as.character(resourcesData$zipcode)
```

```
15
16 # merge with court records data
17 data0812 <- merge(data0812, resourcesData, by.x = "zip", by.y = 'zipcode', all.x = T)
18
19 # compute zipcode income terciles based on the final, defendant-level analysis sample (last
       cases only)
20 data0812.last <- lastCase(data0812)
21 quants <- quantile(data0812.last$avgAGI, probs = seq(0, 1, 1/3), na.rm = T)
22
23 # construct tercile indicators
24 data0812$quants_inc <- NA
25 data0812$quants_inc[data0812$avgAGI > quants[3]] <- 3
26 data0812$quants_inc[data0812$avgAGI > quants[2] & data0812$avgAGI <= quants[3]] <- 2
27 data0812$quants_inc[data0812$avgAGI <= quants[2]] <- 1
```

Listing 3: Income merge