This script repeats the analyses used to initially select PA as a proxy for MA crime rates, substituting the 2014 data used in the original script with new data from 2018. To do so, we calculate Euclidian distance between two-state (MA & all 50 others (inc. DC)) matrices, using both standardized and raw rates of crime.

So first things first, let's load our libraries and data:

```
knitr::opts_chunk$set(warning = F)
library(here) ## relative pathways
```

```
## here() starts at /home/mikemahoney218/codebase/clean-slate/analyses/determine_closest_state_2018_fbi
```

```
library(readxl) ## reading excel
library(dplyr) ## data manipulation
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr) ## nesting dataframes
library(purrr) ## map reduce functions
FBIData <- read_excel("../../data/cleaned/fbi_aggregated_2018/2018_FBI_aggregate_crime_data.xlsx")
```

The first thing that was done with the 2014 data was to calculate the euclidian distance between each state and Massachusetts. Now, we're using a different data source altogether here, so I have to take a bit of a different approach in my wrangling – most importantly, I have to calculate the per-capita arrest rate myself (I'm assuming these numbers aren't per capita or 100k based on California reporting about 1 million arrests for its 40 million population).

```
nestedFBIData <- FBIData %>%
  filter(age_category == "Under 18") %>%
  ## these variables are the same ones done with 2014 data
  select(state,
         year,
         robbery,
         property_crime,
         burglary,
         larceny_theft,
         motor_vehicle_theft,
         estimated_population) %>%
  ## get per-capita crime rate
  mutate_if(is.numeric, funs(. / estimated_population)) %>%
  select(-estimated_population) %>%
  ## this bit is wonky if you don't know R; I'm creating a column of dataframes
  ## containing data for only that state
  nest(data = -c(state)) %>%
  rename(other_states = data)

## this is decidedly not the smartest way to do this, but:
## iterate across each state, combining its dataframe with the MA one;
## create a list of those dataframes in "frames"
frames <- vector("list")
```

```
for (i in seq_along(nestedFBIData$state)) {
  frames[i] <- nest(data = everything(),
                    rbind(nestedFBIData[[i, "other_states"]],
                          nestedFBIData[nestedFBIData$state == "Massachusetts", ][[2]][[1]]))
}

## now iterate through those dataframes calulating euclidian distance
## (same metric as last time)
distScore <- vector()
for (i in seq_along(frames)) {
  distScore[i] <- dist(frames[[i]][[1]])
}
## now let's label those scores and examine the results!
as_tibble(cbind(nestedFBIData$state, distScore)) %>%
  arrange(distScore) %>%
  rename(State = V1)
```

```
## # A tibble: 51 x 2
##    State         distScore
##    <chr>         <chr>
##  1 Massachusetts 0
##  2 New Jersey    0.000188674684947384
##  3 Kentucky      0.000257620022372041
##  4 Michigan      0.0002952224008481
##  5 Washington    0.000314373961025925
##  6 California    0.00031515634078624
##  7 Ohio          0.000320012579670631
##  8 New York      0.000352020022462097
##  9 Virginia      0.000355923271521075
## 10 Pennsylvania  0.000373193483555347
## # ... with 41 more rows
```

So PA in this quick experiment comes in 10th place (if you count MA as 1st; which is... a choice) – which is still 80th percentile, but I'm wondering if we had pragmatic considerations for selecting PA in addition to distance-based ones. I think that, unless it turns out that New Jersey publishes all their crimes in a Google Spreadsheet emailed to everyone on New Year's, we can still justify looking at PA with this result – the distinction between any top 10 state other than NJ and maybe Kentucky is extremely minimal.

Of course, the last analysis standardized the variables we looked at as well, so that rates of offense among less common crimes could be weighted equally to more common offenses. I need to do an inch more reading to have an opinion on this, I think – it seems to me like we should care about the crimes with more absolute cases more here (and weight them accordingly), as our desired outcome isn't as much "what state has the same offender profiles" as it is "what has the same rate of crimes" – that is, I think that a large percentage difference in a more common crime category is more important to us than a similar magnitude but smaller absolute number difference in a less common one. But I'm not entirely sure, so here's the analysis run with standardized data:

```
nestedFBIData <- FBIData %>%
  filter(age_category == "Under 18" &
    state != "Iowa") %>%
  select(state,
         year,
         robbery,
         property_crime,
         burglary,
```

```
         larceny_theft,
         motor_vehicle_theft,
         estimated_population) %>%
  mutate_if(is.numeric, funs(. / estimated_population)) %>%
  mutate_if(is.numeric, scale) %>%
  select(-estimated_population) %>%
  nest(data = -c(state)) %>%
  rename(other_states = data)

frames <- vector("list")
for (i in seq_along(nestedFBIData$state)) {
  frames[i] <- nest(data = everything(),
                    rbind(nestedFBIData[[i, "other_states"]],
                          nestedFBIData[nestedFBIData$state == "Massachusetts", ][[2]][[1]]))
}

## now iterate through those dataframes calulating euclidian distance
## (same metric as last time)
distScore <- vector()
for (i in seq_along(frames)) {
  distScore[i] <- dist(frames[[i]][[1]])
}
## now let's label those scores and examine the results!
as_tibble(cbind(nestedFBIData$state, distScore)) %>%
  arrange(distScore) %>%
  rename(State = V1)
```

```
## # A tibble: 50 x 2
##    State         distScore
##    <chr>         <chr>
##  1 Massachusetts 0
##  2 Vermont       0.07295737902621
##  3 Hawaii        0.300822161968777
##  4 New Hampshire 0.396801185187358
##  5 Kansas        0.522340189932416
##  6 Kentucky      0.734254074928748
##  7 West Virginia 0.767584554749902
##  8 Virginia      0.866830147930307
##  9 New Jersey    0.909700210753361
## 10 Nebraska      1.13043159690538
## # ... with 40 more rows
```

```
as_tibble(cbind(nestedFBIData$state, distScore)) %>%
  arrange(distScore) %>%
  rename(State = V1) %>%
  tail(-10)
```

```
## # A tibble: 40 x 2
##    State        distScore
##    <chr>        <chr>
##  1 Ohio         1.18961576021385
##  2 Pennsylvania 1.46130399733378
##  3 New Mexico   1.47315484582252
##  4 Indiana      1.49349419972769
##  5 Illinois     1.51953805040245
```

```
##  6 Michigan     1.547730594564
##  7 New York     1.64278455807131
##  8 Oregon       1.68122260065554
##  9 Utah         1.8239792819845
## 10 North Dakota 1.97347691869274
## # ... with 30 more rows
```

PA now is number 12, down two spots. Notably, Vermont came from almost dead last to an undisputable second place here. Having Vermont and New Hampshire here makes an amount of sense to me – those are extremely similar states, after all. I'll look forward to talking about this with people on Tuesday – I'm not even clear myself on what this implies for our next steps.