Hate crime's intimidation effects and EEOC filings

Rowan Shi

2019-05-04

A crime is categorized as a hate crime if the victims were targeted due to their affiliation or membership to a specific race, sexual orientation, religion, gender identity, or similar. While some perpetrators of hate crime do not describe an outright explantion of their actions, an FBI investigation on the motivations behind hate crimes found defensive and retaliatory reasons among others. (The others were for thrill-seeking or ideologically symbolic purposes.)

Does hate crime "work"?

Defensive and retalitory motivations suggest an implicit hope that the crime will cause a change in behavior of the minority group. The study characterizes the retaliatory crimes as terrorism, language consistent with this idea. For example, a recent shooter who killed two women in a yoga studio was a self-proclaimed far-right extremist. In a series of online videos, he criticized women for not engaging romantically with him, especially white women in interracial relationships (for betraying "their blood"), as well as for wearing yoga pants in public. In the context of these videos, his shooting could be seen as a warning to women to change their behavior.

This project proposes to study the effect of hate crime on minority behavior. In particular: does hate crime have a qualitatively or quantitatively important effect? Is the effect one the perpetrator may ostensibly hope for?

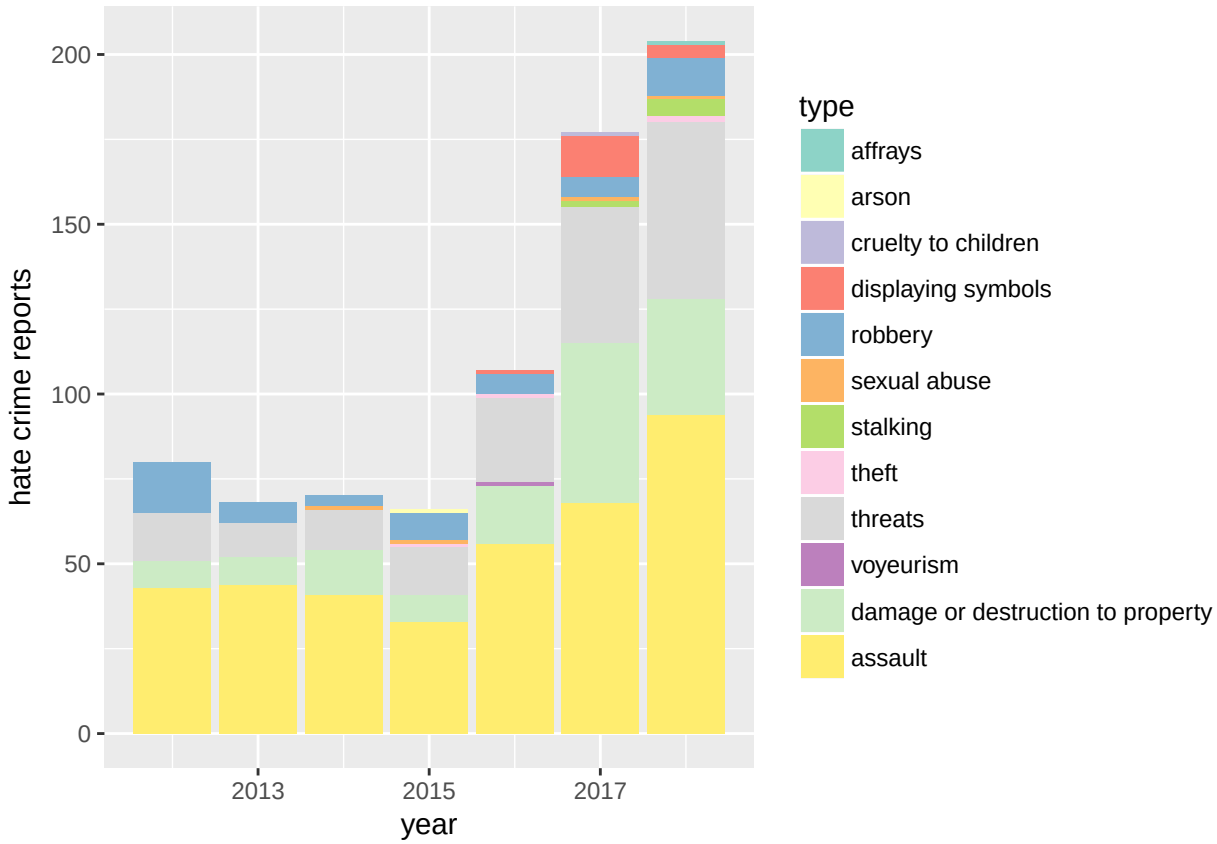
The answers to these questions could have possible applications to policy, but would also be relevant for individuals part of minority groups as well as the services that help them. Additionally, the results may be of interest to companies that value creating a safe work environment for its employees.

D.C.: A case study

Datasets of hate crime reports are becoming available at the city-level, as part of a larger Police Data Initiative. In what follows below, we use data from D.C. for preliminary exploration. Details on data and its processing can be found at the end of the document.

A first look at D.C. hate crime data

The hate crime data extends from 2012 to the present, represented in the chart below.

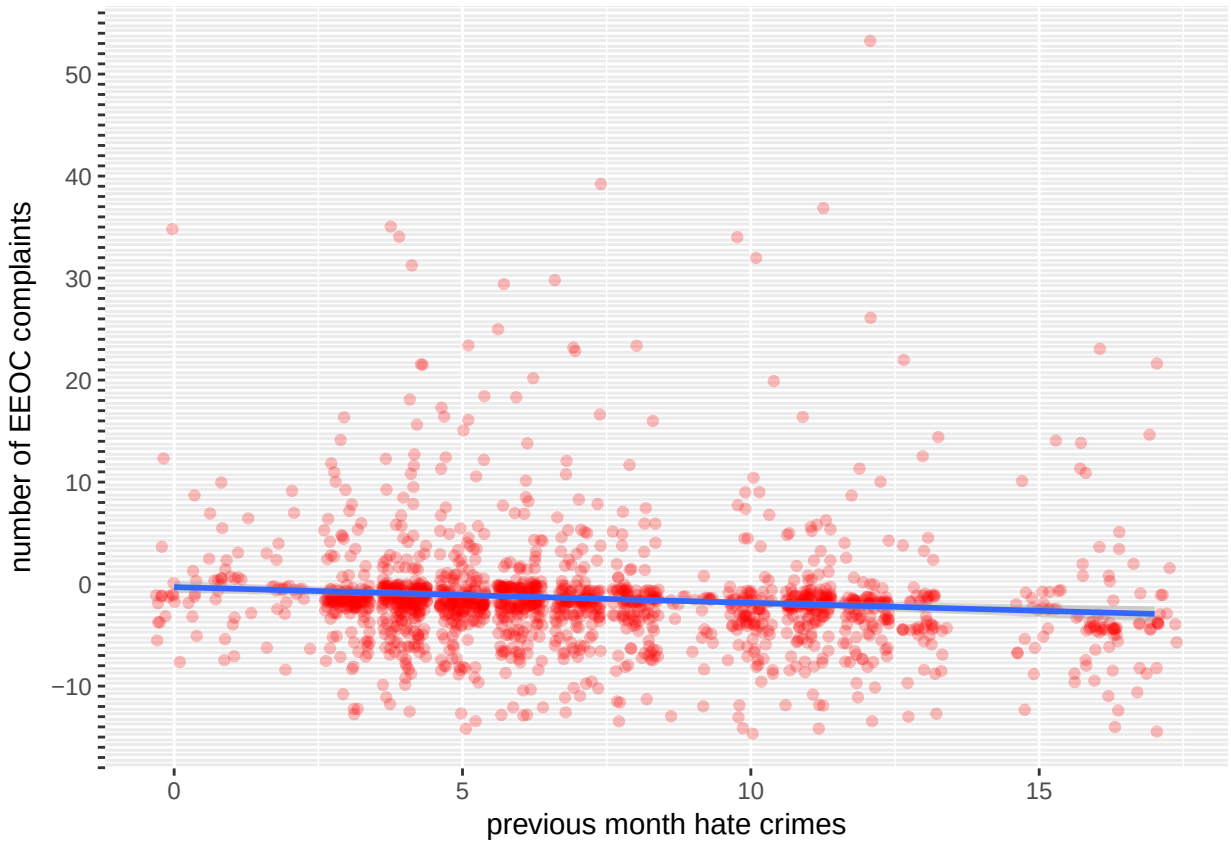


While the incidence of hate crimes was fairly stable between 2012 and 2015, a marked increase has taken place since. In particular, the rates of assault, property damage, and threats have risen. In addition, there has been an emergence of a new type of crime, "displaying symbols". These types of crimes, especially threats, are broadly suggestive of the highlighted motivations above. The trend over time emphasizes the increasing importance of the questions this project addresses.

Preliminary analysis on EEOC reportings

To evaluate the effect of hate crimes, this case study will use data from the EEOC (Equal Employment Opportunity Commission). The measured variable will be the number of discrimination complaints filed with the Commission. If the hate crimes truly have a terrorism-like effect, it may be reasonable to expect that the number of filings may decrease in response to higher rates of hate crime.

Merging in the complaint count data from the EEOC, the chart below plots complaint counts each month (demeaned by year, month, and sector means) against the number of hate crimes from the previous month. The lagged hate crime incidence is used to allow a reaction time to the hate crimes.



The regression (excluding the year, month, and sector fixed effects for readability) associated with the graph is presented below.

```
                   Estimate Std. Error    t value   Pr(>|t|)
hate_count       0.006794903 0.05569245  0.1220076 0.90290866
lag1_hate_count -0.134371059 0.06063901 -2.2159176 0.02684055
lag2_hate_count -0.065442905 0.06165911 -1.0613664 0.28868718
```

With the linear analysis, an additional hate crime in the D.C. area is associated with 0.1343711 fewer reports of discrimination to the EEOC. A median 7 hate crimes per month is associated with 0.9405974 fewer EEOC complaints each month. For context, a median 207 complaints are made to the EEOC in D.C. each month.

A larger project scope

The hate crime and EEOC data also report the specific minority group targeted, which could lead to more precise analysis and possibly larger and more easily interpretable results. In addition, evidence from more cities as well as comparison across cities could add depth to the conclusions.

Preparing the EEOC data

Source

The data is sourced from https://github.com/PublicI/employment-discrimination, where it is contained in two text delimited files: one containing 2010 data and 2011-2017 data. They are loaded into R and appended below.

```
library(data.table)
library(magrittr)
library(stringr)

# prep
## load
y2010 = fread("2010.txt", showProgress = T)
y2010_vars = c("charge_num", "state", "num_employees_code", "num_employees", "naics_code", "naics", "ty
names(y2010) = y2010_vars
rest = fread("2011_2017.txt")
vars = c("year", "charge_num", "state", "num_employees_code", "num_employees", "naics_code", "naics", "
names(rest) = vars
## append
y2010_vars[!(y2010_vars %in% vars)]
vars[!(vars %in% y2010_vars)]
y2010[, c("date_fepa_sent_to_eeoc", "year") := .(NULL, "FY2010")]
(all(names(y2010) %in% names(rest)) & all(names(rest) %in% names(y2010))) %>% assertthat::assert_that()
full = rbind(rest, y2010[, ..vars])
# full[, lapply(.SD, factor)] %>% summary
```

Formatting and cleaning

The data is mainly text data, which is organized and parsed below. There are a few pairs of variables where the first variable is a code and the second variable is the value of the code: for example, `basis_code` contains a letter code corresponding to the description provided in `basis`. These variable pairs are first used to compile dictionaries for the code variables; then the code variables are converted into factors and the description variables are deleted. As an exception, the NAICS variable is left as is, since codes are given at 6-digit precision, but 2- or 4- digit precision may be preferrable for analysis.

```
gen_codebook = function(code_var, label_var) {
  var_pair = c(code_var, label_var)
  codebook = full[, ..var_pair] %>% unique
  codebook = codebook[get(code_var) != "null" & get(code_var) != ""]
  codebook[get(label_var) == "", (label_var) := "Other"]
}
gen_factor = function(i) {
  full[, c(code_vars[[i]], label_vars[[i]]) := .(factor(get(code_vars[[i]]), levels = codebooks[[i]][[1
}
code_vars = grep("code", vars, value = T)
label_vars = str_replace(code_vars, "_code", "")
codebooks = mapply(gen_codebook, code_vars, label_vars, SIMPLIFY = F)
lapply(which(!grepl("naics", code_vars)), gen_factor) # leave naics as is
```

Then, additional string variables with no accompanying description variables are converted into factors.

```
replace_empty_with_na = function(var_name) {
  full[get(var_name) == "", (var_name) := NA]
  full[get(var_name) == "null", (var_name) := NA]
}
other_factor_vars = c("state", "sex", "court", "case_type")
lapply(other_factor_vars, replace_empty_with_na)
full[, c(other_factor_vars) := lapply(.SD, factor), .SDcols = other_factor_vars]
```

The data contains some date variables, which are provided in string format but must be parsed into date

format. Some date strings contain four-digit years while others contain 2-digit years. Ambiguity arises from 2-digit years between 00-17. However, most date variables concern EEOC complaint filings between 2010-2017, so the ambiguous dates for these variables are interpreted as being from the 2000s. However, the `birth_date` variable can be equally unreasonably assigned to teh 1900s or 2000s. Since there are few observations with 2-digit year `birth_date` values between 00-17 are, they are dropped from the dataset.

```
conv_dates = function(date_var) {
  full[grepl("/[0-9]{4}$", get(date_var)), paste0(date_var, "_processed") := as.Date(get(date_var), for
  full[grepl("/[0-9]{2}$", get(date_var)), paste0(date_var, "_processed") := as.Date(get(date_var), for
  full[, paste0(date_var, c("", "_processed")) := .(get(paste0(date_var, "_processed")), NULL)]
}
full = full[!grepl("/0[0-9]$", birth_date) & !grepl("/1[0-7]$", birth_date)] # drop ambiguous '00 - '17
full[grepl("/[0-9]{2}$", birth_date), birth_date := sapply(birth_date, str_replace, "/([0-9]{2}$)", "/19
date_vars = grep("date", vars, value = T)
lapply(date_vars, conv_dates)
full = full[year(birth_date) > 1917 & year(birth_date) < 2010]
```

Finally, numeric variables must be parsed. The data contains commas separating blocks of thousands digits, so these are removed before conversion.

```
full[, c("year") := .(as.integer(str_extract(year, "[0-9]+")))]
number_vars = c("charge_num", "monetary_benefits", "litigation_monetary_benefits")
lapply(number_vars, replace_empty_with_na)
full[, (number_vars) := lapply(.SD, function(x) as.numeric(str_replace_all(x, ",", ""))), .SDcol = numbe
```

Pruning

Some entries in the data are not actual reports since each variable value is just the name of the variable. These are dropped before the final cleaned dataset is saved. Likewise, age and length of case are calculated with the given date variables; observations with ages outside of a reasonable range (18-85) and with invalid case lengths (negative) are dropped.

```
### drop entries that aren't cases
full = full[court != "COURT" | is.na(court)]
full[, naics := NULL]
lapply(c("naics_code", "civil_action_num"), replace_empty_with_na)
full[, c("case_length", "age") := .(difftime(date_closed, date_received, units = "days"), as.integer(fl
full = full[((age >= 18 & age <= 85) | is.na(age)) & ((case_length >= 0) | is.na(case_length))]
```

Generating counts data

Each observation in the data represents one complaint to the EEOC. These are aggregated at the month-state-sector level into counts, adding in zeros for cells for which there are no complaints in the dataset. Sectors retain 2-digit detail for industries, with categorizations such as Agriculture, Manufacturing, and Educational Services. Since the data only extends to September 2017, combinations from October 2017 - December 2017 are not included.

```
eeoc_counts = full[, .(year = year(date_received), month = month(date_received), state, naics2 = factor
setkeyv(eeoc_counts, c("year", "month", "naics2", "state"))
fill_combinations = function(var_list, data) {
  build_scaffold = function(var_list) {
    get_values = function(variable) {
      data[, get(variable)] %>% unique %>% factor
    }
```

```
    values = lapply(var_list, get_values)
    scaffold = do.call(expand.grid, values) %>% setDT
    names(scaffold) = var_list
    setkeyv(scaffold, var_list)
    scaffold
  }
  merge(build_scaffold(var_list), copy(data)[, (var_list) := lapply(.SD, factor), .SDcols = var_list], a
}
eeoc_counts = fill_combinations(c("year", "month", "state", "naics2"), eeoc_counts)
eeoc_counts[is.na(eeoc_count), eeoc_count := 0]
eeoc_counts = eeoc_counts[!(year == 2017 & month %in% 10:12)]
```

Preparing the DC hate crimes data

Source

The hate crimes data is sourced from https://mpdc.dc.gov/node/1334781, as part of the larger Police Data Initiative. It is loaded into R below.

```
dc_hate = fread("dc_hate.csv")
names(dc_hate) = c("offense_date", "offense_time", "report_date", "report_year", "report_month", "ccn",
```

Formatting and cleaning

The data contains mostly string information, which is parsed into dates and factors below.

```
## dates
dc_hate[, offense_date := as.Date(offense_date, format = "%m/%d/%Y")]
dc_hate[, report_date := as.Date(report_date, format = "%m/%d/%y")]
dc_hate[, c("report_year", "report_month") := NULL]
## string/factors
replace_empty_with_na = function(var_name) {
  dc_hate[get(var_name) == "", (var_name) := NA]
}
factor_vars = c("bias", "targeted_group", "offense")
lapply(factor_vars, replace_empty_with_na)
dc_hate[, (factor_vars) := lapply(.SD, function(x) factor(tolower(x))), .SDcols = factor_vars]
```

Generating counts data

```
dc_hate_counts = dc_hate[, .(year = year(offense_date), month = month(offense_date))][, .(hate_count =
dc_hate_counts = fill_combinations(c("year","month"), dc_hate_counts)
dc_hate_counts[is.na(hate_count), hate_count := 0]
```