Project 2: Gun Violence

TOTAL: 20pts

PURPOSE

To understand patterns of gun violence across U.S. cities and use those patterns to understand similarities and differences to Washington, D.C.

GOAL

Analyze U.S. police shooting event data across behavior, space, and time.

DATA

Law enforcement shootings collected and aggregated by The Washington Post. Why? Because the government doesn't keep national-level records, unfortunately.

METHODS

Step 0: Load and install packages

1. You will need to install these packages (run both of these if using the AU computers):
    a. install.packages('tidyverse')
    b. install.packages('leaflet')
    c. install.packages('tidycensus')
    d. install.packages(c("cowplot", "ggrepel", "rgeos", "sf", "maps", "usmap", "ggspatial", "libwgeom", "rnaturalearth", "rnaturalearthdata"))
2. You will need to load these libraries, no matter the computer:
    a. library(tidyverse)
    b. library(leaflet)
    c. library(tidycensus)
    d. library(rnaturalearth)
    e. library(rnaturalearthdata)
    f. library(maps)
    g. library(lubridate)
    h. library(usmap)

i. library(sf)
j. theme_set(theme_bw())

Step 1: Get data

1. Read in your file, using the read.csv function:
   - wapo.data <- read.csv("https://raw.githubusercontent.com/washingtonpost/data-police-shootings/master/v2/fatal-police-shootings-data.csv", stringsAsFactors = FALSE)
2. Welp, that was easy.
3. Open the table, and explore the fields available.

Step 2: Clean data

Using the event table from Step 1…

1. Convert the date to a date format:
   - wapo.data$date <- as.Date(wapo.data$date)
2. Create a Month field
   a. wapo.data$month <- substr(wapo.data$date, 6, 7)
3. Create a Year field
   a. wapo.data$year <- substr(wapo.data$date, 0, 4)
4. Calculate a combination of Year and Month
   a. wapo.data$yearmonth <- paste(wapo.data$year, wapo.data$month, sep = "-")
5. Calculate a combination of State and City
   a. wapo.data$statecity <- paste(wapo.data$state, wapo.data$city, sep = "-")
6. Adjust race?
   a. unique(wapo.data$race)
   b. wapo.data$race <- gsub("W;B;N", "O", wapo.data$race)
   c. wapo.data$race <- gsub("N;H", "O", wapo.data$race)
   d. wapo.data$race <- gsub("W;H", "O", wapo.data$race)
   e. wapo.data$race <- gsub("B;H", "O", wapo.data$race)
   f. wapo.data$race <- gsub("W;B", "O", wapo.data$race)
   g. wapo.data$race <- gsub("W;A", "O", wapo.data$race)
7. Check the location data, and *potentially* get rid of blank and trash locations:
   a. wapo.data.map <- subset(wapo.data, !is.na(wapo.data$latitude))
8. Success!

Step 3: Summarize data

Next, summarize the events by year, by month, and by state, as well as calculate some statistics.

In this project, we need to summarize the events by numerous fields. So, four steps:

1. Summarize by race (or any other variable):
   a. wapo.race <- wapo.data %>%
      group_by(race) %>%

```
summarise(count = n()) %>%
mutate(pct = round(count/sum(count)*100, 2))
```

b. Click on the table 'wapo.race' on the top right pane and you'll see counts by race

2. Repeat the process (the single line of code above):
    i. City
    ii. Year
    iii. Armed
    iv. Signs of Mental Illness
    v. Flee
    vi. Gender?
    vii. Body camera?
3. Create multi-value summaries:
    - By Race and Manner of Death
        i. `wapo.race.mental <- wapo.data %>% group_by(race, was_mental_illness_related) %>% summarise(count = n()) %>% mutate(pct = count/sum(count)*100)`
    - This combines state, city, and race
        i. `wapo.race.city <- wapo.data %>%`
        ii. `group_by(statecity, race) %>%`
        iii. `summarise(count = n()) %>%`
        iv. `mutate(pct = count/sum(count)*100)`
    - Repeat for Race and…
        i. Armed
        ii. State
        iii. Mental Illness
        iv. Threat Level
        v. Flee
        vi. Body Camera
        vii. Year
        viii. Year-Month

Step 4: Visualize data

First, try the explore package… maybe.

Second, make some bar graphs. Some examples:

- Events by Race
    - `ggplot(wapo.data) + geom_bar(aes(x = race), stat = "count", fill = "grey")`
- Events by Mental Illness and Race
    - `ggplot(wapo.race.mental, aes(x = factor(race), y = pct, fill = factor(was_mental_illness_related))) + geom_bar(stat="identity", width = 0.7) + labs(x = "Race", y = "percent", fill = "Mental Illness Related") + theme_minimal(base_size = 14)`
    - `ggplot(wapo.data) + geom_bar(aes(x = was_mental_illness_related), stat = "count", fill = "grey") + facet_wrap(~ race, nrow = 3)`
- Events by Armed and City
    - `sum.city.armed <- wapo.data %>%`

```
    group_by(statecity, armed_with) %>%
    summarise(count = n()) %>%
    mutate(pct = round(count/sum(count)*100,2))

ggplot(sum.city.armed, aes(x = statecity, y = count, fill =
                           factor(armed_with))) + geom_bar (stat = "identity") +
  labs(x = "State-City", y = "Number of Police Shootings",
       title = "Police Shootings by Victim Weapon Type", subtitle = "XXX", fill = "Victims
Weapon") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

- Experiment with other variables
- An example:
  - Sum events by City and Armed_with
    ```
    wapo.city.armed <- wapo.data %>%
      group_by(statecity, armed_with) %>%
      summarise(count = n()) %>%
      mutate(pct = count/sum(count)*100)
    ```
  - Clean/filter select weapons
    ```
    wapo.city.weapon <- subset(wapo.city.armed, armed_with %in% c("unarmed",
    "gun", "blunt_object"))
    ```
  - Filter select cities
    ```
    wapo.city.select <- subset(wapo.city.weapon, statecity %in%
                               c("DC-Washington","FL-Orlando","MI-Detroit","AZ-Glendale",
                                 "TX-Corpus Christi","KS-Wichita","MO-Springfield"))
    ```
  - Graph
    ```
    ggplot(wapo.city.select, aes(x = statecity, y = count, fill = factor(armed_with))) +
      geom_bar (stat = "identity") +
      labs(x = "State-City", y = "Number of Police Shootings", title = "Police Shootings
    by Victim Weapon Type",
           subtitle = "XXX", fill = "Victims Weapon") +
      theme(axis.text.x = element_text(angle = 45, hjust = 1))
    ```

Third, make some maps:

1. Create layers for the US/World, and the US states:
   - `world <- ne_countries(scale = "medium", returnclass = "sf") # this builds a list of countries`
   - `states <- st_as_sf(map("state", plot = FALSE, fill = TRUE)) # this cleans up the US states`
2. Make a transparent point pin map:
   i. `ggplot(data = world) + geom_sf() + geom_sf(data = states, fill = NA) + geom_point(data = wapo.data, aes(x = longitude, y = latitude), size = 2, alpha = 0.05) + coord_sf(xlim = c(-135, -60), ylim = c(25, 50), expand = FALSE)`
   ii. ggplot creates the base map
   iii. geom_point creates the points
      1. Alter the size, shape, fill, and alpha parameters to change the style
   iv. coord_sf is the zoom level
   v. Notice the missing rows! Those events would be the ones occurring in Hawaii and Alaska
3. Facet maps:

- ggplot(data = world) + geom_sf() + geom_sf(data = states, fill = NA) + geom_point(data = wapo.data, aes(x = longitude, y = latitude), size = 2, alpha = 0.05) + coord_sf(xlim = c(-135, -60), ylim = c(25, 50), expand = FALSE) + facet_wrap(~ year, nrow = 2)
    i. ~ year is the parameter to make maps from; experiment with others
    ii. nrow is the number of rows you want for the maps

4. A dynamic, interactive map
    - leaflet(wapo.data) %>%
    - addTiles() %>%
    - addMarkers(lng = ~longitude, lat = ~latitude, clusterOptions = markerClusterOptions())

HERE IS THE SCRIPT FROM CLASS ON 12 FEBRUARY 2025

Step 5: Fuse census data

Join the census data to your events...in ten steps.

1. First, get an API key:
    a. Go here
    b. Sign-up. Check your email. Get your API key.
2. Second, install and load this package and your API key:
    a. census_api_key("YOUR API KEY GOES HERE", install = TRUE, overwrite = TRUE) # run once!
        i. If you experience API discomfort, use this:

    b. **Now you must run this line!**
        i. readRenviron("~/.Renviron")
3. Third, load a list of all the variables to work with:
    a. census.variables.2023 <- load_variables(2023, "acs5", cache = TRUE)
4. Fourth, query some race statistics
    a. race.2023 <- get_acs(geography = "state", variables = c("B02008_001", "B02009_001", "B02010_001", "B02011_001", "B03001_003"), year = 2023)
        i. race.2023 is the object
        ii. get_acs is the function from tidycensus for querying the ACS survey from 2023
        iii. The geography is for each state's statistics
        iv. The variables are the codes for White, Black, Native American, Asian, and Hispanic
        v. The year is 2023
    b. Let's replace the codes with names:
        i. race.2023$variable <- gsub("B02008_001", "White", race.2023$variable)
        ii. Repeat the process for each code and label:
            1. B02009_001 is Black
            2. B02010_001 is Native American
            3. B02011_001 is Asian
            4. B03001_003 is Hispanic
                a. race.2023$variable <- gsub("B02009_001", "Black", race.2023$variable)
                b. race.2023$variable <- gsub("B02010_001", "Native American", race.2023$variable)

c. race.2023$variable <- gsub("B02011_001", "Asian", race.2023$variable)
d. race.2023$variable <- gsub("B03001_003", "Hispanic", race.2023$variable)

c. Let's do a total US race query, too:
i. race.total <- get_acs(geography = "us", variables = c("B02008_001", "B02009_001", "B02010_001", "B02011_001", "B03001_003"), year = 2023)
1. Repeat the process to substitute codes for names
a. race.total$variable <- gsub("B02008_001", "White", race.total$variable)
b. Repeat for the rest

5. Fifth, query some population statistics:
a. totalpop.2023 <- get_acs(geography = "state", variables = "B01003_001", year = 2023)
b. Rename the columns:
i. names(totalpop.2023) <- c("GEOID", "State", "Variable", "Population", "junk")
c. Calculate the percentage of the total population for each state:
i. totalpop.2023$State.PCT <- round(totalpop.2023$Population/sum(totalpop.2023$Population)*100, 2)
6. Sixth, join the total population to the race table:
a. temp.race.population <- race.2023 %>% left_join(totalpop.2023, by = "GEOID")
i. left_join merges the columns from the population table to the race table, using the 'GEOID' as the common field
b. Only keep the relevant columns:
i. race.population <- temp.race.population[c(1:4,8,10)]
c. Rename them:
i. names(race.population) <- c("GEOID", "State", "Race", "Population.Race", "Population.State", "State.PCT")
d. And then calculate the percentage of the total population for each race:
i. race.population$Race.PCT <- round(race.population$Population.Race/race.population$Population.State*100, 2)
7. Seventh, add a percentage to the nationwide race table:
a. race.total$PCT <- race.total$estimate/sum(totalpop.2023$Population)*100
i. Notice, we are using the sum of the state populations from another table to calculate the values for this table
b. Rename the columns:
i. names(race.total) <- c("GEOID", "Area", "Race", "Count", "moe", "Race.PCT")
8. Eighth, clean up the shooting event race and state summary tables:
a. First, the race table:
i. sum.race$race <- gsub("W", "White", sum.race$race)
ii. Repeat for the other races
1. sum.race$race <- gsub("B", "Black", sum.race$race)
2. sum.race$race <- gsub("A", "Asian", sum.race$race)
3. sum.race$race <- gsub("N", "Native American", sum.race$race)
4. sum.race$race <- gsub("H", "Hispanic", sum.race$race)
5. sum.race$race <- gsub("O", "Other", sum.race$race)
b. Rename the columns:
i. names(sum.race) <- c("Race", "Count", "Shooting.PCT")

c. Second, the state table:
   i. Change the abbreviations to full names
      1. sum.state <- wapo.data %>%
      2. group_by(state) %>%
      3. summarise(count = n()) %>%
      4. mutate(pct = round(count/sum(count)*100,2))
      5. sum.state$State.Name <- state.name[match(sum.state$state,state.abb)]
   ii. Fix DC!
      1. sum.state$State.Name <- replace_na(sum.state$State.Name, "District of Columbia")
   iii. Rename the columns:
      1. names(sum.state) <- c("oldstate", "Count", "Shooting.PCT", "State")

9. Ninth, join the race tables
   a. wapo.census.race <- sum.race %>% left_join(race.total, by = "Race")
      i. Remove unnecessary columns
      ii. Rename the remaining columns
         1. wapo.census.race <- wapo.census.race[c(1:3,6,8)]
         2. names(wapo.census.race) <- c("Race", "Shooting.Count", "Shooting.PCT",
         3. "Population.Count", "Race.PCT")
10. Tenth, join the state tables
   a. wapo.census.state <- sum.state %>% left_join(totalpop.2023, by = "State")
      i. Remove unnecessary columns
      ii. Rename the remaining columns
11. Success!

HERE IS THE SCRIPT FROM CLASS ON 19 FEBRUARY 2025

Step 6: Identify similar cities

- Explore the summary tables, graphs, and maps you created for **events** and **victims** and **locations**, for **totals** and **trends over time**
   a. **Events**: the number of police shootings
   b. **Victims**: available attributes (race, gender, weapon, etc.)
   c. **Locations**: the cities, states, and areas (hotspots, coldspots)
      i. Consider similarity based on being in the same hotspot, or being in a hot/cold spot at all
   d. **Totals** vs **trends over time**

ANALYSIS

When conducting your analysis, consider this perspective:

1. State a finding
   a. This is the new thing that you've identified from the data
2. Provide your evidence
   a. This is typically a statistical reference

3. Add the context
    a. This is the integration of the stat(s) and the finding
    b. Compare your finding to the norm/average, or a similar measure in the data

Your goal, for this project:

1. **Five similar cities**
    a. Actual cities, not states
2. **Analyze events**
    a. This is counts of shootings
    b. Analyze these are *counts* and *percentages*
        i. Percentages refers to the percentage of total victims in a city that have specific characteristics
        ii. Compare the percentages to D.C. for similarity
    c. Characterize *yearly* trends for events and victims in D.C. as either ***increasing***, ***decreasing***, or remaining ***stable***
    d. Compare these trends to other cities with similar overall counts
        i. If the overall counts are generally the same, but the trends are different, those cities are not similar
3. **Analyze victims**
    a. This can be any combination of the available attributes, including Race, Sex, Mental Illness, Flee Status, Weapon, etc.
    b. Analyze these are *counts* and *percentages*
        i. Percentages refers to the percentage of total victims in a city that have specific characteristics
        ii. Compare the percentages to D.C. for similarity
4. **Analyze locations**
    a. Make maps
    b. Characterize any potential similar city as either being in the same hotspot as D.C., a hotspot in general, or not in a hotspot
        i. If a potential similar city is not located in a police shooting hotspot, it's not similar to D.C.
5. **One visual per city**
    a. You don't *need* to include maps as your visuals

FORMAT

Submit your script as a usable HTML file, via RMarkdown. You create this output as a .Rmd file in RStudio, but only submit the .html file.

Use this file as a template.

SUBMISSION

Once your analysis is complete, please submit your project as either an HTML file (as the output/export/knit of your RMarkdown script), via Canvas.

GRADES

Data: **2pt**

- Provide a brief paragraph (3-4 sentences) on the data used in this project. This includes the specific source(s), the size, and the specific temporal and spatial constraints.

Methods: **2pt**

- Provide a brief paragraph (3-4 sentences) on the research methods leveraged to answer this question. This includes the software, calculations, skills, techniques, and unique workflows used to analyze your data and develop an answer. You do NOT need to describe click-by-click instructions or lines of code describing how you did things; you DO need to describe a logical process that is specific enough that a reader could replicate.

Analysis: **10pts**

- Using the data specific to this project, identify five cities *similar* to Washington, D.C. (2pts each)
    - Provide justification of similarity for each city (4-5 sentences each)
        - Each justification should include at least two specific reasons for similarity
            - Each reason should include a relevant statistical reference

Visuals: **5pts**

- Create one visual per similar city (1pt each)
- Each visual should have a direct connection to the analysis, and be clearly labeled

Formatting: **1pt**

- Error-free writing. No typos, run-on sentences, or sentence fragments. Proper punctuation. Real words.
    - Need help paraphrasing dense content? Try this. And here's a great reference, too.
- Project created in RMarkdown and submitted as an HTML file via Canvas.

Please email me with any questions.