Project 2: Mapping

TOTAL: 10pts

GOAL

Make a map using Washington, DC crime data.

Step 0: Prep

Install and load packages:

- install.packages('tidyverse')
- install.packages('leaflet')
- install.packages('sf')
- install.packages('tigris')
- install.packages('data.table')
- library(tidyverse)
- library(leaflet)
- library(sf)
- library(tigris)
- library(data.table)

Step 1: Get data

Write a function to grab all the datas:

- We know the data can be downloaded directly from the source from a link like this
- Make an object for the base URL
    - url <- "https://opendata.dc.gov/datasets/DCGIS::crime-incidents-in-"
- Make a list of the years we need to capture
    - years <- c(2008:2025)
- Build the URLs to query
    - full.urls <- paste0(url, years, ".csv")
- Create an empty data table
    - dc.data <- data.frame()
- Create a “for” loop that reads each yearly dataset into R and merges them together

    - for(file in full.urls) {
    -   tmp.data <- read.csv(file, stringsAsFactors = FALSE)
    -   dc.data <- rbind(tmp.data, dc.data)
    - }
- Success.

Step 2: Clean data

Separate the datetime field into 'date' and 'time':

- dc.data <- separate(dc.data, REPORT_DAT, into = c("DATE", "TIME"), sep = " ")

Format the date column as a date:

- dc.data$DATE <- as.Date(dc.data$DATE, format = "%Y/%m/%d")

Step 3: Enrich data

Calculate year, month, day of year, day of week, and hour of day:

- dc.data$YEAR <- substr(dc.data$DATE, 0, 4)
- dc.data$MONTH <- month(dc.data$DATE)
- dc.data$MONTHS <- months(dc.data$DATE)
- dc.data$DAY <- day(dc.data$DATE)
- dc.data$DOW <- weekdays(dc.data$DATE)
- dc.data$HOUR <- substr(dc.data$TIME, 0, 2)

Step 4: Map data

Get rid of any crimes without location data:

- dc.data.full <- subset(dc.data, !is.na(dc.data$LATITUDE))
- And then remove the old data from your environment, to save RAM:
    - rm(dc.data)

Geographic data:

- Get roads, major roads, and a DC boundary outline:
    - dc.roads <- roads("DC", "District of Columbia")
    - dc.roads.major <- dc.roads %>% filter(RTTYP %in% c("I","S","U"))
    - dc.outline <- county_subdivisions("DC", "District of Columbia")
- Get landmarks, and filter out AU:
    - dc.placenames <- landmarks("DC")

  - dc.au <- subset(dc.placenames, dc.placenames$POINTID == "xxx")
- Get water, too:
  - dc.water <- area_water("DC", "District of Columbia")
  - dc.water <- filter(dc.water, AWATER >= 1000)

Empty map of just roads:

- ggplot(dc.roads) + geom_sf()

Points, one color:

- ggplot(dc.data.full, aes(x=LONGITUDE, y=LATITUDE, color = "red")) +
- geom_point()

Points with a city outline and a clean background:

- ggplot() +
- geom_sf(data = dc.outline) +
- geom_point(aes(x=LONGITUDE, y=LATITUDE), color = "red", data = dc.data) +
- theme_void()

Points with roads, a clean background, and some titles:

- ggplot() +
- geom_point(aes(x=LONGITUDE, y=LATITUDE), color = "red", data = dc.data) +
- geom_sf(data = dc.roads.major) +
- theme_void() +
- theme(plot.title = element_text(size = 20, hjust=.5), plot.subtitle = element_text(size = 8, hjust=.5, margin=margin(2, 0, 5, 0))) +
- labs(title = "A Title", subtitle = "Some other text")

Points by year:

- ggplot(dc.data, aes(x=LONGITUDE, y=LATITUDE, color = YEAR)) +
- geom_point()

Points for a day of the week, colored by year:

- ggplot(subset(dc.data, dc.data$DOW == 'Friday'), aes(x=LONGITUDE, y=LATITUDE, color = YEAR)) + geom_point()

Points for a specific year, colored by day of week:

- ggplot(subset(dc.data, dc.data$YEAR == '2023'), aes(x=LONGITUDE, y=LATITUDE, color = DOW)) + geom_point()

Transparent points:

```
ggplot() +
  geom_point(aes(x = LONGITUDE, y = LATITUDE), data = dc.data, alpha = 0.005, size = 0.5) +
  theme(legend.position="bottom")
```

An example:

```
dc1 <- subset(dc.data, dc.data$OFFENSE == 'HOMICIDE')

ggplot() +
  geom_sf(data = dc.outline) +
  geom_point(aes(x=LONGITUDE, y=LATITUDE), color = "red", data = dc1, size = 1.0,
             alpha = 0.15) +
  geom_sf(data = dc.au) +
  geom_sf(data = dc.roads.major) +
  theme_void() +
  theme(plot.title = element_text(size = 20, hjust=.5),
        plot.subtitle = element_text(size = 8, hjust=.5, margin=margin(2, 0, 5, 0))) +
  labs(title = "A Title", subtitle = "Some other text")
```

Density plot:

```
dc2 <- subset(dc.data, dc.data$OFFENSE == 'BURGLARY')

ggplot() +
  stat_density2d(aes(x = LONGITUDE, y = LATITUDE, fill = ..level.., alpha = 0.01),
                 size = 0.01, bins = 50, data = dc.data, geom = "polygon")
```

Contour plot:

```
ggplot() +
  geom_sf(data = dc.outline) +
  geom_density2d(data = dc.data, aes(x = LONGITUDE, y = LATITUDE), size = 0.15)
```

Densities and contours:

```
ggplot() +
  stat_density2d(aes(x = LONGITUDE, y = LATITUDE, fill = ..level.., alpha = 0.01),
                 size = 0.001, bins = 10, data = dc.data, geom = "polygon") +
  geom_density2d(data = dc.data, aes(x = LONGITUDE, y = LATITUDE), size = 0.15)
```

Hexagons:

```
ggplot() +
  geom_hex(aes(x = LONGITUDE, y = LATITUDE), data = xxx, bins = xxx) +
  scale_fill_continuous(type = "viridis")
```

Transparent points per year, or per hour:

- ggplot() + geom_point(aes(x = LONGITUDE, y = LATITUDE), data = dc.data, alpha = 0.01, size = 0.5) + facet_wrap(~ year, nrow = 4)
- ggplot() + geom_point(aes(x = LONGITUDE, y = LATITUDE), data = dc.data, alpha = 0.01, size = 0.5) + facet_wrap(~ hour, nrow = 6)

Transparent points per year, with streets, titles, and a clean background:

- ggplot() +
- geom_sf(data = dc.roads, inherit.aes = FALSE, color = "grey", size = .5, alpha = .6) +
- geom_point(aes(x = LONGITUDE, y = LATITUDE, color = "red"), data = dc.data, alpha = 0.1, size = 0.75) + theme_void() + theme(plot.title = element_text(size = 20, hjust=.5), plot.subtitle = element_text(size = 8, hjust=.5, margin=margin(2, 0, 5, 0))) +
- labs(title = "A Title", subtitle = "With more text here") +
- facet_wrap(~ year, nrow = 5)

Transparent points with landmarks and clean formatting:

- ggplot() +
- geom_sf(data = dc.outline$geometry, fill = "transparent", linewidth = 1,
- color = "black") +
- geom_sf(data = dc.roads$geometry, alpha = 0.1) +
- geom_sf(data = dc.water$geometry, fill = "lightblue") +
- geom_point(data = filter(dc.data, OFFENSE == "HOMICIDE"),
- aes(x = LONGITUDE, y = LATITUDE), size = 1.0, alpha = 0.15, color = "red") +
- geom_sf_text(data = dc.placenames$geometry, label = dc.placenames$FULLNAME,
- check_overlap = TRUE, size = 1.5) +
- theme(plot.title = element_text(hjust = 0.5, size = 14),
- plot.subtitle = element_text(size = 8, hjust=.5, margin=margin(2, 0, 5, 0)),
- axis.text = element_blank(),
- axis.ticks = element_blank(),
- axis.title = element_blank(),
- panel.grid.major = element_blank(),
- panel.background = element_blank()) +
- labs(title = "A Title", subtitle = "Some other text")

Density with landmarks and clean formatting:

- ggplot() +
- geom_sf(data = dc.outline$geometry, fill = "transparent", linewidth = 1,
- color = "black") +
- geom_sf(data = dc.roads$geometry, alpha = 0.1) +
- geom_sf(data = dc.water$geometry, fill = "lightblue") +
- stat_density2d(data = filter(dc.data, OFFENSE == "BURGLARY"),
- aes(x = LONGITUDE, y = LATITUDE, fill = ..level..),
- bins = 10, h = 0.01, geom = "polygon", alpha = 0.75) +

```
geom_sf_text(data = dc.placenames$geometry, label = dc.placenames$FULLNAME,
        check_overlap = TRUE, size = 1.5) +
scale_fill_viridis_c(option = "plasma") +
theme(plot.title = element_text(hjust = 0.5, size = 14),
    axis.text = element_blank(),
    axis.ticks = element_blank(),
    axis.title = element_blank(),
    panel.grid.major = element_blank(),
    panel.background = element_blank())
```

Interactive clusters - basic:

```
leaflet(dc.data) %>%
  addTiles() %>%
  addMarkers(lng = ~LONGITUDE, lat = ~LATITUDE, popup = dc.data$OFFENSE,
        clusterOptions = markerClusterOptions())
```

Interactive clusters - new basemap:

```
leaflet(dc.data) %>%
  addProviderTiles("CartoDB.DarkMatter") %>%
  addMarkers(lng = ~LONGITUDE, lat = ~LATITUDE, popup = dc.data$OFFENSE,
        clusterOptions = markerClusterOptions())
```

Interactive clusters - city boundary and more pop-up text:

```
dc3 <- subset(dc.data, dc.data$OFFENSE == 'ROBBERY')

leaflet(dc3) %>%
  addTiles() %>%
  addMarkers(lng = ~LONGITUDE, lat = ~LATITUDE,
        popup = paste(
          "Crime Type: ", dc3$OFFENSE, "<br>",
          "Date:", dc3$DATE, "<br>",
          "Shift: ", dc3$SHIFT),
        clusterOptions = markerClusterOptions()) %>%
  addPolygons(data = dc.outline)
```

THIS IS THE SCRIPT WE BUILT IN CLASS ON 05 FEBRUARY 2025

HERE IS THE SCRIPT WE BUILT IN CLASS ON 12 FEBRUARY 2025

Part II: SUBMISSION

Transform your script as a usable HTML file, via RMarkdown. You will want to copy/paste the most relevant parts of the script you've built into a clean, consolidated file. You create this output as a .Rmd file in RStudio, but what you submit will be an .HTML file.

Use this file as a template

1. Adjust the template with your code accordingly
2. Use the 'knit' button in RStudio to produce the output
    a. The .HTML is automatically saved to the same folder as your script

HERE IS THE TEMPLATE WE MADE IN CLASS ON 12 FEBRUARY 2025

GRADES

- **Part I: Graph (9pts)**
    - Create three maps using the data provided
    - Each map should analyze a different variable in the data (**3pts each**)
        - *One map should be a transparent point map based on a subset of crime(s)*
        - *One map should be a density/contour/hex map based on a second subset of crime(s)*
        - *One map should be an interactive cluster map based on a third subset of crime(s)*
    - Each map should be in its own section, clearly identifying what crime subsets are being visualized
- **Part II: Submission (1pt)**
    - Project created in RMarkdown and submitted as an HTML file via Canvas

Please email me with any questions.