

Homework 5

Ainsley Lotito

November 12, 2025

Looking at the `data` subdirectory

Write a piece of code that lists the files you saved in the `data` subdirectory.

```
list.files("../data")  
  
## [1] "homicide-data.csv"
```

Mapping a city

Pick one city in the data. Create a map showing the locations of the homicides in that city, using the `sf` framework discussed in class. Use `tigris` to download boundaries for some sub-city geography (e.g., tracts, block groups, county subdivisions) to show as a layer underneath the points showing homicides. Use different facets for solved versus unsolved homicides and different colors to show the three race groups with the highest number of homicides for that city (you may find the `fct_lump` function from `forcats` useful for this).

Importing data

To start off, let's read in the data and take a look at the cities we may choose from.

```
library(tidyverse)  
library(forcats)  
library(tigris)  
library(labelled)  
  
homicides <- read.csv("../data/homicide-data.csv")  
unique(homicides$city)
```

I think it would be fun to use Baton Rouge, so I am going to create an object containing just the observations for Baton Rouge.

```
baton_rouge <- homicides %>%  
  filter(city == "Baton Rouge")  
dim(baton_rouge)  
  
## [1] 424 12
```

```

baton_rouge %>%
  slice(1:3)

## # A tibble: 3 x 9
##   uid reported_date victim_last victim_first victim_race victim_ag
##   <chr> <date>        <chr>       <chr>      <chr>      <chr>
## 1 Bat-000001 20110103 GREVIOUS TERRANCE Black    3
## 2 Bat-000002 20110212 ERVIN      CALVIN     Black    4
## 3 Bat-000003 20110212 WATSON     BRANDON    Black    3
## # A tibble: 3 x 6
##   victim_sex city state      lat      lon disposition
##   <chr>       <chr> <chr> <dbl> <dbl> <chr>
## 1 Male        Baton Rouge LA  30.52505 -91.18410 Open/No arrest
## 2 Male        Baton Rouge LA  30.48617 -91.14824 Open/No arrest
## 3 Male        Baton Rouge LA  30.48617 -91.14824 Open/No arrest

## # A tibble: 5 x 5
##   Asian    Black Hispanic Other  White
##   <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     1     375     11      2     35

## # A tibble: 2 x 2
##   Female Male
##   <dbl> <dbl>
## 1    67    357

## # A tibble: 10 x 10
##    0     1     14    15    16    17    18    19    2
##    6     1     2     2     4     7     19    20    1
##   21    22    23    24    25    26    27    28    29
##   16    23    28    18    15     8    16    21    14
##   31    32    33    34    35    36    37    38    39
##   11    15     8    11     8     7     6    10     6
##   41    42    43    44    45    46    47    48    49
##    5     3     7     1     8     5     4     1     7
##   50    51    52    53    54    55    56    57    59
##    3     6     6     3     4     2     4     5     3
##   61    62    63    64    68    69     7    70    71
##    2     1     3     1     1     1     1     1     2
##   76    77 Unknown
##    1     1     1

## [1] "Open/No arrest"          "Closed by arrest"         "Closed without arrest"

```

I have data on 424 homicides in Baton Rouge, with 12 columns which include identifying id, victim name, race, age, sex), the latitude and longitude, and the disposition. I need cleaning to re-categorize some variables, and to determine whether the case is solved vs. unsolved.

```
baton_rouge <- baton_rouge %>%  
  mutate(reported_date = ymd(reported_date),  
  
         case_status = case_when(  
           disposition == "Open/No arrest" ~ "Solved",  
           disposition %in% c("Closed by arrest", "Closed without arrest")  
             ~ "Unsolved"),  
         case_status = factor(case_status),  
  
         victim_age = victim.age %>%
```

```

    na_if("Unknown") %>%
      as.numeric(),

    victim_sex = fct_relevel(victim_sex,
                             "Male", "Female"), #Ordering by count

    victim_race = fct_relevel(victim_race,
                             "Black", "White", "Hispanic", "Other", "Asian") #Ordering by count
  )
baton_rouge %>%
  slice(1:3)

##          uid reported_date victim_last victim_first victim_race victim_age
## 1 Bat-000001    2011-01-03   GREVIOUS     TERRANCE     Black       38
## 2 Bat-000002    2011-02-12     ERVIN       CALVIN     Black       46
## 3 Bat-000003    2011-02-12    WATSON      BRANDON     Black       30
##   victim_sex      city state      lat      lon disposition case_status
## 1     Male Baton Rouge     LA 30.52505 -91.18410 Open/No arrest     Solved
## 2     Male Baton Rouge     LA 30.48617 -91.14824 Open/No arrest     Solved
## 3     Male Baton Rouge     LA 30.48617 -91.14824 Open/No arrest     Solved

```

Lastly, I need save as an sf object so I can plot this on a map.

```

#changing to an sf object
library(sf)
baton_rouge_sf <- st_as_sf(baton_rouge, coords = c("lon", "lat")) %>%
  st_set_crs(4269)
baton_rouge_sf %>% slice(1:3)
dim(baton_rouge_sf)

```

Making a map of baton rouge. I need to pull some tracts, block groups, and county subdivisions from tigris. The FIPS code for East Baton Rouge Parish is 22033, and West is 22121. I can therefore select by the countyFP of 033 and 121. I am going to start by just making a base map of Baton Rouge before I add my homicide data.

```

library(tigris)
la_tracts <- tracts(state = "LA") #tracts first
ba_tracts <- la_tracts %>%
  filter(COUNTYFP %in% c("033", "121"))

la_blocks <- block_groups(state = "LA")
ba_blocks <- la_blocks %>%
  filter(COUNTYFP %in% c("033", "121"))

ba_subdivisions <- county_subdivisions("LA", county = c("033", "121"))

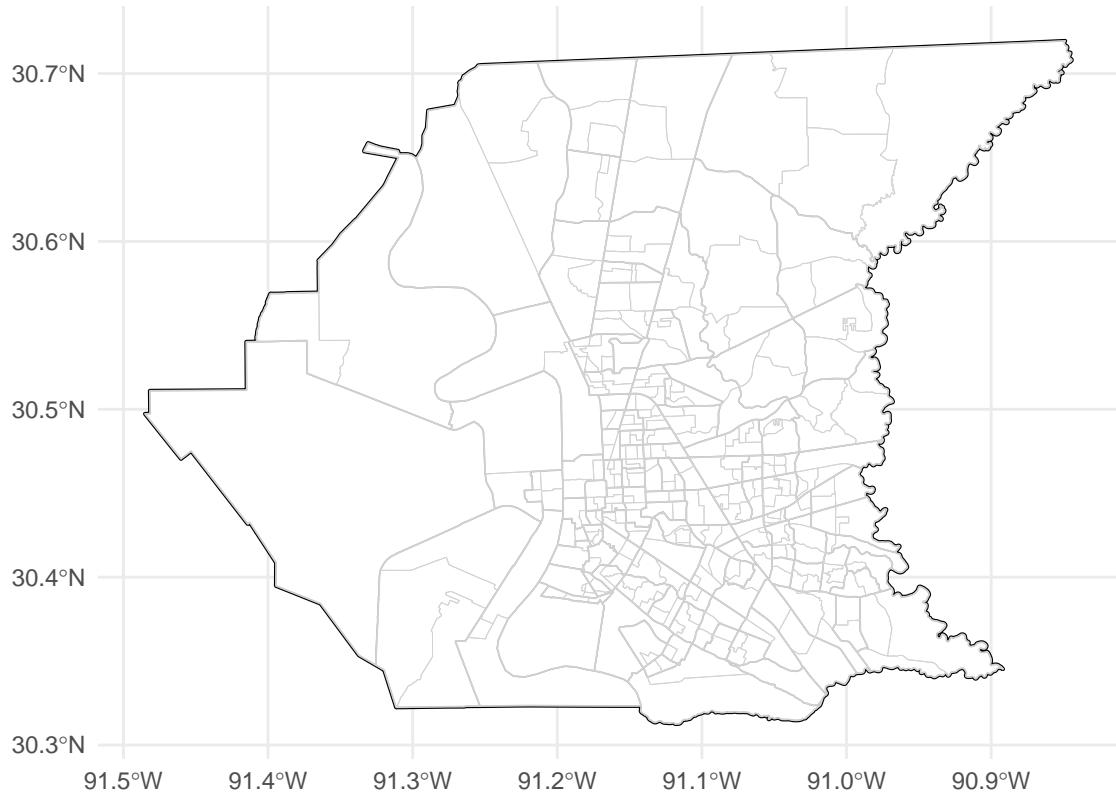
```

```

ggplot() +
  geom_sf(data = ba_subdivisions, fill = NA, color = "black", size = 0.7) + #subdivisions
  geom_sf(data = ba_tracts, fill = "white", color = "gray", size = 0.3) + #tracts
  geom_sf(data = ba_blocks, fill = NA, color = "lightgray", size = 0.1) + #blocks
  theme_minimal() +
  labs(title = "Baton Rouge, Louisiana")

```

Baton Rouge, Louisiana

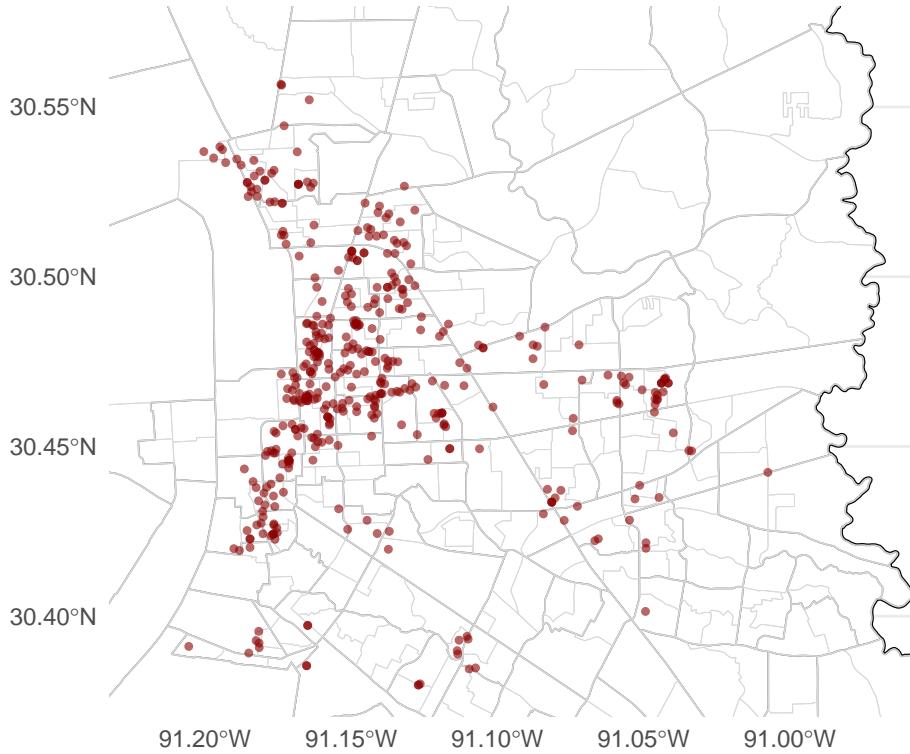


Now that I have a basemap, I can look into adding some homicide info. The first map is going to just show the locations of each homicide.

```
ggplot() +
  geom_sf(data = ba_subdivisions, fill = NA, color = "black", size = 0.7) + #subdivisions
  geom_sf(data = ba_tracts, fill = "white", color = "gray60", size = 0.3) + #tracts
  geom_sf(data = ba_blocks, fill = NA, color = "gray85", size = 0.1) + #blocks
  geom_sf(data = baton_rouge_sf, color = "darkred", size = 0.9, alpha = .6) +
  theme_minimal() +
  xlim(c(-91.22, -90.97)) + ylim(c(30.38, 30.57)) + #zooming in to points
  labs(title = "Homicides in Baton Rouge, LA",
       subtitle = "Jan 3, 2011 - Dec 31, 2017")
```

Homicides in Baton Rouge, LA

Jan 3, 2011 – Dec 31, 2017



Next we are going to look at solved versus unsolved homicides, and use different colors for the three race groups with the highest number of homicides. I also got some errors and found that I will need to use the command inherit.aes = FALSE for each sf layer without the case_status facetting variable (all the geography ones)

```
top3_races <- baton_rouge_sf %>%
  group_by(victim_race) %>%
  count() %>%
  slice(1:3)

baton_rouge_sf_top3 <- baton_rouge_sf %>%
  filter(victim_race %in% top3_races$victim_race) %>%
  set_variable_labels(
    victim_race = "Racial Group of Victim"
  )

ggplot() +
  geom_sf(data = ba_subdivisions, fill = NA, color = "black", size = 0.7,
         inherit.aes = FALSE) + #subdivisions
  geom_sf(data = ba_tracts, fill = "white", color = "gray60", size = 0.3,
         inherit.aes = FALSE) + #tracts
  geom_sf(data = ba_blocks, fill = NA, color = "gray85", size = 0.1,
         inherit.aes = FALSE) + #blocks

  geom_sf(data = baton_rouge_sf_top3, aes(color = victim_race), size = 1.1, alpha = .6) +
```

```

facet_wrap(~ case_status) +
theme_minimal() +
xlim(c(-91.22, -90.97)) + ylim(c(30.38, 30.57)) + #zooming in to points
labs(title = "Homicides in Baton Rouge 2011 – 2017",
     subtitle = "By Racial Group and Case Status") +
theme(plot.title = element_text(hjust = 0.5),
      plot.subtitle = element_text(hjust = 0.5))

```

