Homework5_Markdown

Madison Russell

2022-11-30

Packages needed for this homework assignment:

```
library(readr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v dplyr   1.0.9
## v tibble  3.1.8     v stringr 1.4.1
## v tidyr   1.2.1     v forcats 0.5.2
## v purrr   0.3.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(tidyverse)
library(sf)
```

```
## Linking to GEOS 3.10.2, GDAL 3.4.2, PROJ 8.2.1; sf_use_s2() is TRUE
```

```
library(tigris)
```

```
## To enable caching of data, set ‘options(tigris_use_cache = TRUE)‘
## in your R script or .Rprofile.
```

```
library(ggplot2)
library(purrr)
library(scales)
```

```
##
## Attaching package: ’scales’
##
## The following object is masked from ’package:purrr’:
##
##     discard
##
## The following object is masked from ’package:readr’:
##
##     col_factor
```

```
library(ggthemes)
library(broom)
library(forcats)
library(geofacet)
library(tidyr)
library(dplyr)
```

Choice 1: 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).

Let's read in the original data set, make a Milwaukee "tracts" sf, and filter out the original data set to create a new object called "milwaukee_homicides".

```
homicides_url <- paste0("https://raw.githubusercontent.com/washingtonpost/",
                           "data-homicides/master/homicide-data.csv")
homicides <- read.csv(homicides_url)
milwaukee_tracts <- tracts("WI", "Milwaukee",
cb = TRUE, class = "sf") %>%
st_as_sf(milwaukee_tracts, coords = c("lon", "lat")) %>%
st_set_crs(4269)
```

```
## Retrieving data for the year 2020
```

```
##   |                                                                            |
```

```
class(milwaukee_tracts)
```

```
## [1] "sf"         "data.frame"
```

```
class(milwaukee_tracts$geometry)
```

```
## [1] "sfc_MULTIPOLYGON" "sfc"
```

```
milwaukee_homicides <- homicides %>%
filter(city == "Milwaukee") %>%
select(victim_race, uid, city, disposition, lat, lon) %>%
dplyr::mutate(victim_race = forcats::fct_lump(victim_race, n = 3)) %>%
mutate(unsolved_homicides = as.numeric(disposition != "Closed by arrest"))
```

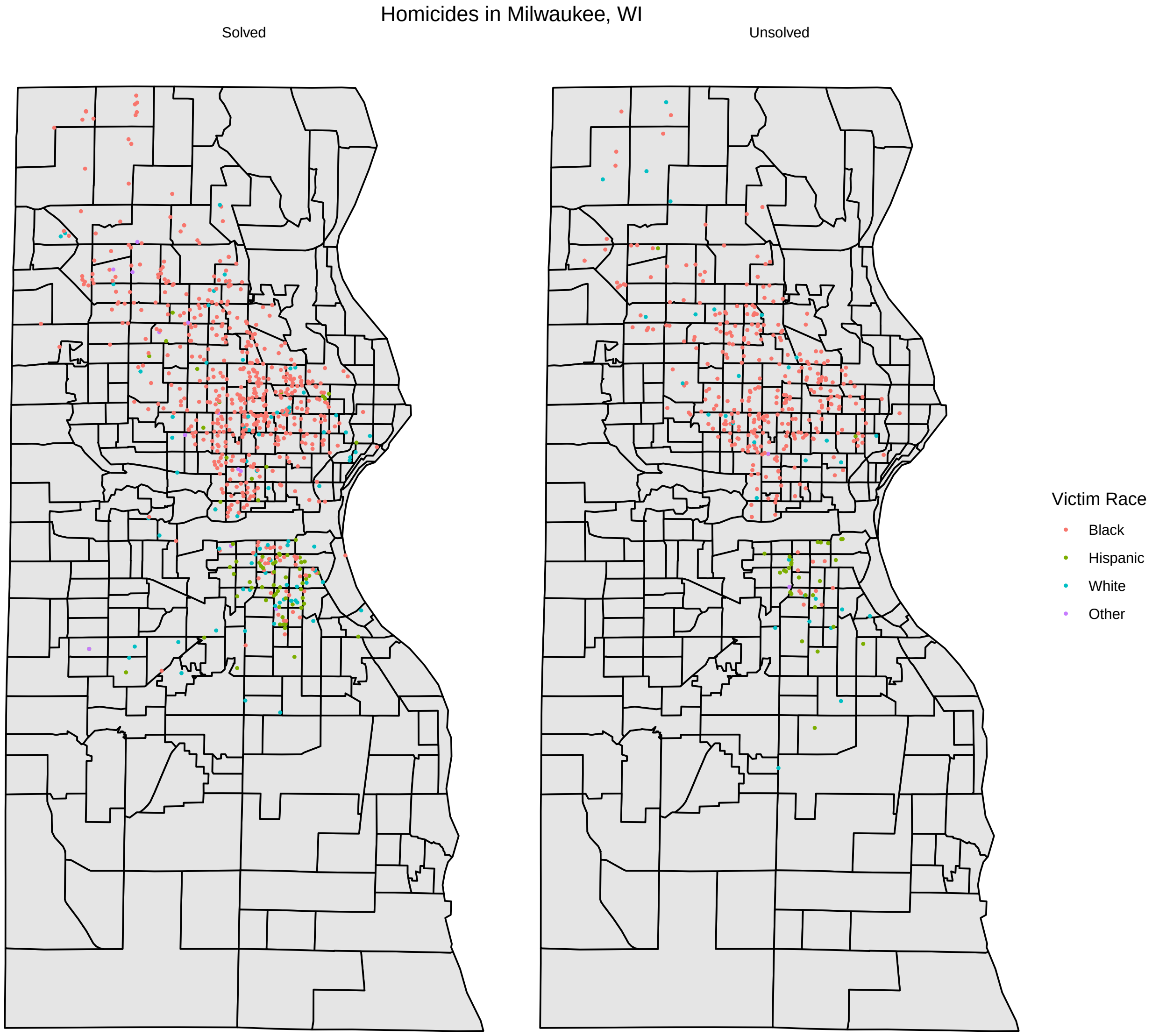
Now that we have our two new objects, let's map, facet, and make our final plot :)

```
plottybaby2 <- ggplot() +
geom_sf(data = milwaukee_tracts, color = "black") +
geom_point(data = milwaukee_homicides, aes(x = lon, y = lat,
    col = victim_race), size = .52) + labs(x = "Longitude",
    y = "Latitude", col = "Victim Race") +
  labs(x = "Longitude", y = "Latitude", col = "Victim Race") + theme_void()

milwaukee_labels <- as_labeller(c('0' = "Solved",
                                  '1' = "Unsolved"))

final_plotty_baby <- plottybaby2 + facet_wrap(.~unsolved_homicides,
  ncol = 2, labeller = milwaukee_labels) +
ggtitle("Homicides in Milwaukee, WI") +
theme(plot.title = element_text(hjust = .5))

final_plotty_baby
```


Homicides in Milwaukee, WI
Solved
Unsolved
Victim Race
Black
Hispanic
White
Other