

PBL Draft

Andy, Blair, Julien

2026-02-27

Library Downloads

```
library(tidyverse)
library(sf)
library(units)
library(readr)
library(tmap)
library(viridis)
library(dplyr)

library(tidycensus)
library(tigris)
options(tigris_use_cache = TRUE)
CRS.new <- st_crs("EPSG:3435")
```

We are limiting this analysis to the year 2025. No way to actually filter CTA Rail data before download.

Data Retrieval

```
crimes_raw <- st_read("crimes_2025.csv")
```

```
## Reading layer 'crimes_2025' from data source
##   'C:\Users\enigh\OneDrive\Desktop\SOSC 13220\Final Project\crimes_2025.csv'
##   using driver 'CSV'
```

```
## Warning: no simple feature geometries present: returning a data.frame or tbl_df
```

```
cta_rail_stations <- st_read("cta_rail_stations.csv")
```

```
## Reading layer 'cta_rail_stations' from data source
##   'C:\Users\enigh\OneDrive\Desktop\SOSC 13220\Final Project\cta_rail_stations.csv'
##   using driver 'CSV'
```

```
## Warning: no simple feature geometries present: returning a data.frame or tbl_df
```

```
police_station_raw <- st_read("police_stations.csv")
```

```
## Reading layer 'police_stations' from data source  
## 'C:\Users\enigh\OneDrive\Desktop\SOSC 13220\Final Project\police_stations.csv'  
## using driver 'CSV'
```

```
## Warning: no simple feature geometries present: returning a data.frame or tbl_df
```

```
cta_ridership_raw <- st_read("cta_rail_ridership.csv")
```

```
## Reading layer 'cta_rail_ridership' from data source  
## 'C:\Users\enigh\OneDrive\Desktop\SOSC 13220\Final Project\cta_rail_ridership.csv'  
## using driver 'CSV'
```

```
## Warning: no simple feature geometries present: returning a data.frame or tbl_df
```

Load in polygon dataset:

```
com_areas <- st_read("comm_area.geojson")
```

```
## Reading layer 'comm_area' from data source  
## 'C:\Users\enigh\OneDrive\Desktop\SOSC 13220\Final Project\comm_area.geojson'  
## using driver 'GeoJSON'  
## Simple feature collection with 77 features and 9 fields  
## Geometry type: MULTIPOLYGON  
## Dimension: XY  
## Bounding box: xmin: -87.94011 ymin: 41.64454 xmax: -87.52414 ymax: 42.02304  
## Geodetic CRS: WGS 84
```

Data Cleaning/Wrangling

We need to filter CTA ridership data to just the year 2025. We also need to aggregate data, as ridership is currently broken down into months.

```
cta_ridership_clean <- cta_ridership_raw %>%  
  mutate(YEAR = as.integer(YEAR)) %>%  
  filter(YEAR == 2025)  
  
cta_ridership_clean <- cta_ridership_clean %>%  
  mutate(TOTAL_RIDES = as.numeric(TOTAL_RIDES)) %>% # ensure numeric  
  group_by(RIDERSHIP_ID, NAME) %>%  
  summarise(  
    total_rides_2025 = sum(TOTAL_RIDES, na.rm = TRUE),  
    .groups = "drop"  
  ) %>%  
  arrange(desc(total_rides_2025))  
  
head(cta_ridership_clean)
```

```
## # A tibble: 6 x 3
##   RIDERSHIP_ID NAME          total_rides_2025
##   <chr>         <chr>          <dbl>
## 1 1660         Lake/State          3212890
## 2 890          O'Hare Airport      3005653
## 3 380          Clark/Lake          2798412
## 4 1220         Fullerton           2370481
## 5 1450         Chicago/State       2363911
## 6 260          State/Lake          2337971
```

```
head(crimes_raw)
```

```
##           ID Case.Number          Date          Block IUCR
## 1 14120097    JK160527 01/01/2026 12:00:00 AM 030XX W FRANKLIN BLVD 0810
## 2 14118983    JK159390 01/01/2026 12:00:00 AM 053XX S MICHIGAN AVE 1305
## 3 14112519    JK151612 01/01/2026 12:00:00 AM 102XX S EMERALD AVE 1565
## 4 14103838    JK140954 01/01/2026 12:00:00 AM 064XX S STEWART AVE 0810
## 5 14100237    JK136833 01/01/2026 12:00:00 AM 042XX W 31ST ST 0486
## 6 14096413    JK131515 01/01/2026 12:00:00 AM 006XX N LOCKWOOD AVE 1130
##           Primary.Type          Description Location.Description
## 1           THEFT              OVER $500          APARTMENT
## 2    CRIMINAL DAMAGE          CRIMINAL DEFACEMENT    OTHER (SPECIFY)
## 3           SEX OFFENSE INDECENT SOLICITATION OF A CHILD    RESIDENCE
## 4           THEFT              OVER $500          APARTMENT
## 5           BATTERY          DOMESTIC BATTERY SIMPLE    RESIDENCE
## 6 DECEPTIVE PRACTICE          FRAUD OR CONFIDENCE GAME    APARTMENT
## Arrest Domestic Beat District Ward Community.Area FBI.Code X.Coordinate
## 1 false    false 1221      012    27              23      06
## 2 false    false 0225      002     3              40      14      1178064
## 3 false    false 2232      022    21              73      17      1173114
## 4 false    true  0722      007     6              68      06      1174731
## 5 false    true  1031      010    22              30     08B     1148654
## 6 false    false 1524      015    37              25      11      1140914
## Y.Coordinate Year          Updated.On          Latitude          Longitude
## 1              2026 02/25/2026 03:43:24 PM
## 2    1869625 2026 02/25/2026 03:41:59 PM 41.797566426 -87.62254141
## 3    1837048 2026 02/18/2026 03:55:47 PM 41.708281913 -87.641654132
## 4    1862261 2026 02/09/2026 03:40:49 PM 41.777433807 -87.63498329
## 5    1883704 2026 02/05/2026 03:40:58 PM 41.836817925 -87.730030636
## 6    1903732 2026 02/05/2026 03:40:58 PM 41.891923124 -87.757939633
##           Location
## 1
## 2 (41.797566426, -87.62254141)
## 3 (41.708281913, -87.641654132)
## 4 (41.777433807, -87.63498329)
## 5 (41.836817925, -87.730030636)
## 6 (41.891923124, -87.757939633)
```

We are also going to ignore crimes entries that have no location data (as otherwise, it is not possible to gauge whether it occurred within a buffer or not.) We also see that the datasets have latitude and longitude data but not point. We will convert to point data as well.

Crimes:

```

crimes_clean <- crimes_raw %>% select(-Location)

crimes_clean <- crimes_clean %>%
  mutate(Latitude = as.numeric(Latitude), Longitude = as.numeric(Longitude))

crimes_clean <- crimes_clean %>% filter(!is.na(Latitude) & !is.na(Longitude)) # we go from 236549 obser

crimes_sf <- crimes_clean %>%
  st_as_sf(coords = c("Longitude", "Latitude"), crs = 4326, remove = FALSE)

```

Ridership:

```

cta_rail_stations_sf <- cta_rail_stations %>%
  st_as_sf(wkt = "the_geom", crs = 4326)

```

Police stations:

```

police_station_clean <- police_station_raw %>% select(-LOCATION)

police_station_clean <- police_station_clean %>%
  mutate(LATITUDE = as.numeric(LATITUDE), LONGITUDE = as.numeric(LONGITUDE))

police_station_clean <- police_station_clean %>% filter(!is.na(LATITUDE) & !is.na(LONGITUDE)) # we go f

police_station_sf <- police_station_clean %>%
  st_as_sf(coords = c("LONGITUDE", "LATITUDE"), crs = 4326, remove = FALSE)

```

Transform to all use same CRS.

```

com_areas <- st_transform(com_areas, CRS.new)
crimes_sf <- st_transform(crimes_sf, CRS.new)
cta_rail_stations_sf <- st_transform(cta_rail_stations_sf, CRS.new)
police_station_sf <- st_transform(police_station_sf, CRS.new)

```

ESDA

First, plotting police stations + CTA rail stations.

```

tmap_mode("plot")

tm_shape(com_areas) + tm_borders() +
  tm_shape(police_station_sf) + tm_symbols(fill = "blue", shape = 24, size = 0.5) +
  tm_shape(cta_rail_stations_sf) + tm_symbols(fill = "red", size = 0.5) +
tm_layout(
  legend.outside = TRUE,
  legend.outside.position = "right",
  outer.margins = c(0.02, 0.02, 0.02, 0.18)) +
tm_layout(
  main.title = paste0("Resource Points"),
  main.title.size = 1.5,
  main.title.position = c("center", "top"),

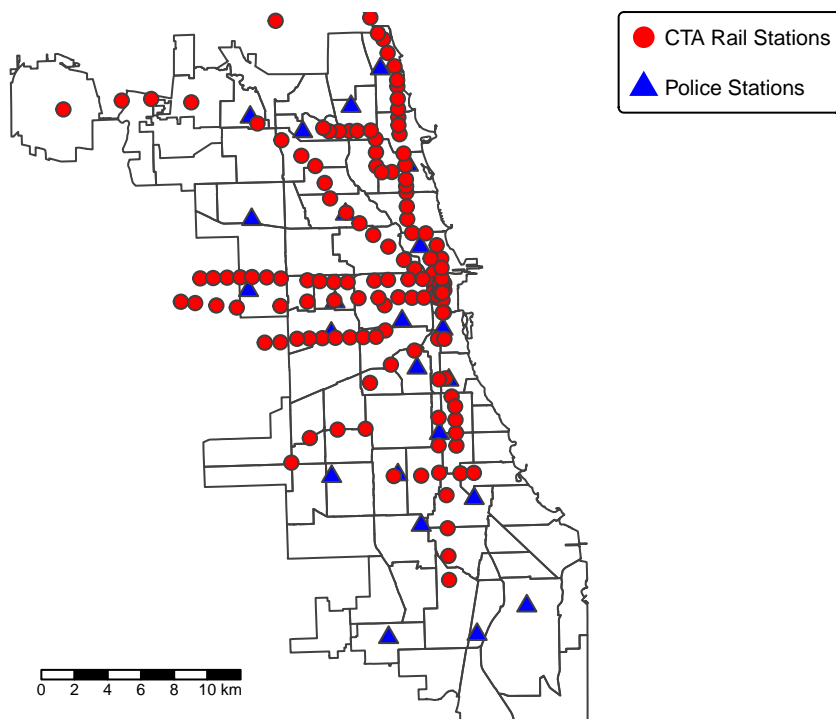
```

```

legend.outside = TRUE,
bg.color = "white",
frame = FALSE
) +
tm_scalebar(position = c("left", "bottom"),
text.size = 0.5 ) +
tm_add_legend(
type = "symbol",
labels = "CTA Rail Stations",
col = "red",
shape = 16,
size = 0.75
) +
tm_add_legend(
type = "symbol",
labels = "Police Stations",
col = "blue",
shape = 17,
size = 0.75
)

```

Resource Points



Buffers:

```

buffer_quarter_mi <- st_buffer(cta_rail_stations_sf, 0.25 * 5280)
buffer_quarter_mi <- st_transform(buffer_quarter_mi, CRS.new)

```

```

buffer_half_mi <- st_buffer(cta_rail_stations_sf, 0.5 * 5280)
buffer_half_mi <- st_transform(buffer_half_mi, CRS.new)

buffer_1_mi <- st_buffer(cta_rail_stations_sf, 5280)
buffer_1_mi <- st_transform(buffer_1_mi, CRS.new)

```

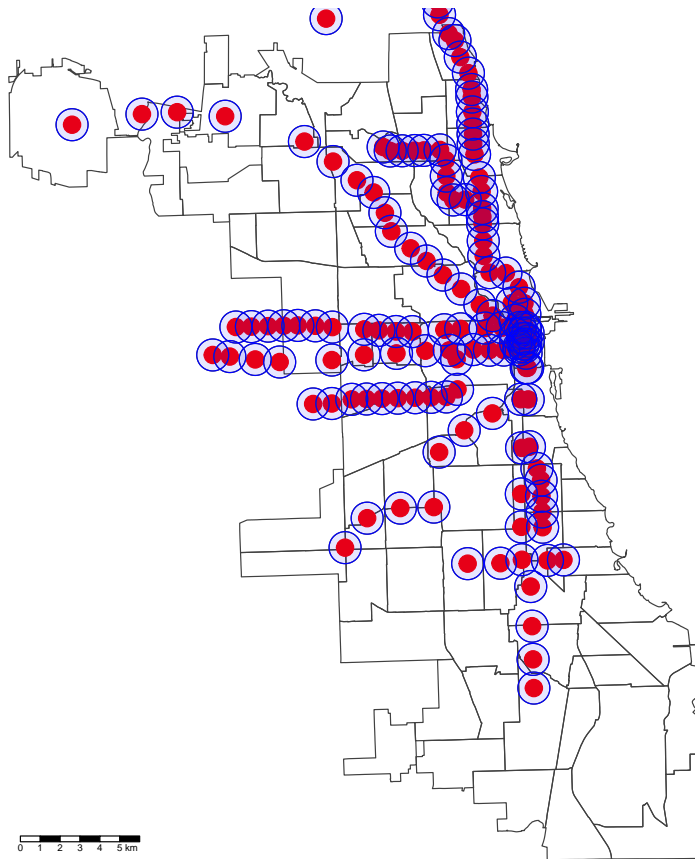
Buffer plots:

```

tmap_mode("plot")
tm_shape(com_areas) +
  tm_borders() +
  tm_shape(cta_rail_stations_sf) +
  tm_dots(fill="red", size=1) +
  tm_shape(buffer_half_mi) +
  tm_fill(col = "blue", alpha = 0.1) +
  tm_borders(col = "blue") +
  tm_layout(
    main.title = paste0("CTA Rail Stations: 0.25-mile Buffer"),
    main.title.size = 1.5,
    main.title.position = c("center", "top"),
    legend.outside = TRUE,
    bg.color = "white",
    frame = FALSE
  ) +
  tm_scalebar(position = c("left", "bottom"),
    text.size = 0.5 )

```

CTA Rail Stations: 0.25-mile Buffer



Note: Do we want to get rid of the CTA Rail stations that are part of the network but technically outside of Chicago?

Crime count in buffer:

```
crime_025_buffer <- lengths(st_intersects(buffer_quarter_mi, crimes_sf))
crime_05_buffer <- lengths(st_intersects(buffer_half_mi, crimes_sf))
crime_1_buffer <- lengths(st_intersects(buffer_1_mi, crimes_sf))

cta_rail_stations_sf$crime_count_025 <- crime_025_buffer
cta_rail_stations_sf$crime_count_05 <- crime_05_buffer
cta_rail_stations_sf$crime_count_1 <- crime_1_buffer

#IMPORTANT: overlapping buffers count crime twice
```

Crime Exposure Visualization:

```
cta_rail_stations_sf <- cta_rail_stations_sf %>%
  mutate(
    crime_level_025 = case_when(
      crime_count_025 >= quantile(crime_count_025, 0.75, na.rm = TRUE) ~ "High",
      crime_count_025 >= quantile(crime_count_025, 0.25, na.rm = TRUE) ~ "Medium",
      TRUE ~ "Low"
    ),
    crime_level_025 = factor(crime_level_025, levels = c("Low", "Medium", "High"))
  )

exposure_025_buffer <- st_buffer(cta_rail_stations_sf, 0.25 * 5280)
```

```
cta_rail_stations_sf <- cta_rail_stations_sf %>%
  mutate(
    crime_level_05 = case_when(
      crime_count_05 >= quantile(crime_count_05, 0.75, na.rm = TRUE) ~ "High",
      crime_count_05 >= quantile(crime_count_05, 0.25, na.rm = TRUE) ~ "Medium",
      TRUE ~ "Low"
    ),
    crime_level_05 = factor(crime_level_05, levels = c("Low", "Medium", "High"))
  )

exposure_05_buffer <- st_buffer(cta_rail_stations_sf, 0.5 * 5280)
```

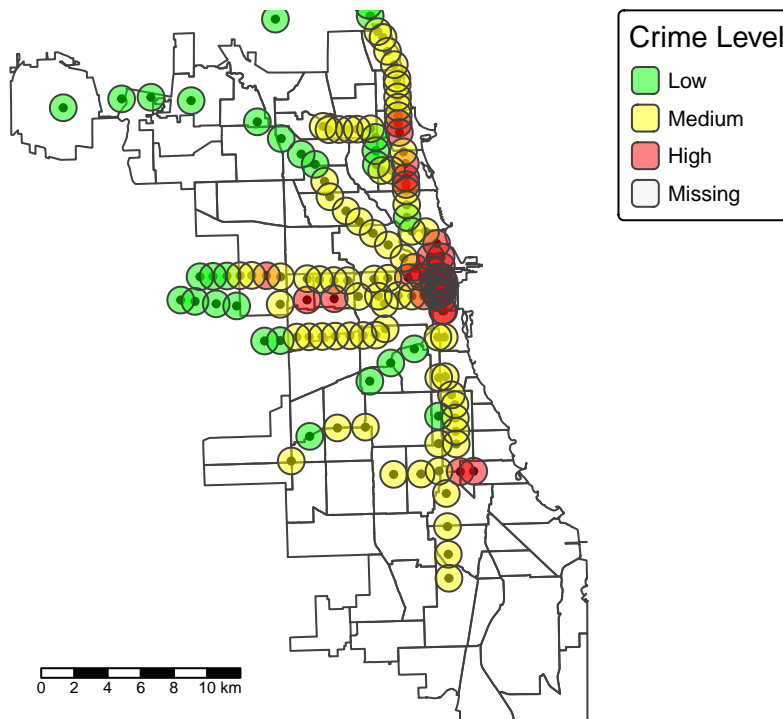
```
tmap_mode("plot")
tm_shape(com_areas) +
  tm_borders() +
  tm_shape(cta_rail_stations_sf) +
  tm_dots(fill="black", size=0.25) +
  tm_shape(exposure_05_buffer) +
  tm_fill(
    col = "crime_level_05",
    palette = c("green", "yellow", "red"),
    title = "Crime Level",
    border.col = NA,
    alpha = 0.5
```

```

) +
tm_layout(
  main.title = paste0("CTA Rail Stations: 0.5-mile Buffer"),
  main.title.size = 1.5,
  main.title.position = c("center", "top"),
  legend.outside = TRUE,
  bg.color = "white",
  frame = FALSE
) +
tm_scalebar(position = c("left", "bottom"),
  text.size = 0.5 )

```

CTA Rail Stations: 0.5-mile Buffer



Next, we want to go into ridership levels. Idea: also colorcode like above, with varying point colors.