Geospatial Analysis in R

Kim Kreiss

Introduction

Brief re-intro to spatial data and mapping

Kernel Density Estimation

Global and Local Moran's I

Getis-Ord / G*

Wrap-up

Introduction

Why R?

- Reproducibility and Transparency
- Flexibility and Customization
- Integration with Statistical Analysis
- Geospatial Visualization

Today

- We will analyze crime data in Chicago together:
  - Combine point data and boundary file
  - Generate informative visualizations
  - Apply statistical techniques to identify hot and cold spots
- This is an interactive workshop, where you are provided with code and data to run locally:
  - `01-geospatial-workshop.Rmd`
  - `2016-2020-chicago-crime.csv`
  - `Boundaries - Neighborhoods.geojson`
  - `Boundaries - Census Tracts - 2010.geojson`

Brief re-intro to spatial data and mapping

Spatial Data

- Most commonly point data, boundary data, grid or raster data
  - point data: longitude and latitude of a unit of interest
  - boundary data: polygon/shape data delineating a geographical area
  - grid or raster data: usually squares/rectangles over the Earth's surface with specific attributes
- Uses Coordinate Reference System (CRS)

Mapping

- Common application is to show point data within a region or geographic boundaries of interest, ie, are homicides in Chicago concentrated in certain neighborhoods?
- Let's map crime in Chicago neighborhoods and census tracts

```
# Load required libraries
library(sf)
library(spdep)
library(spatstat)
library(tidyverse)
library(colorspace)
```

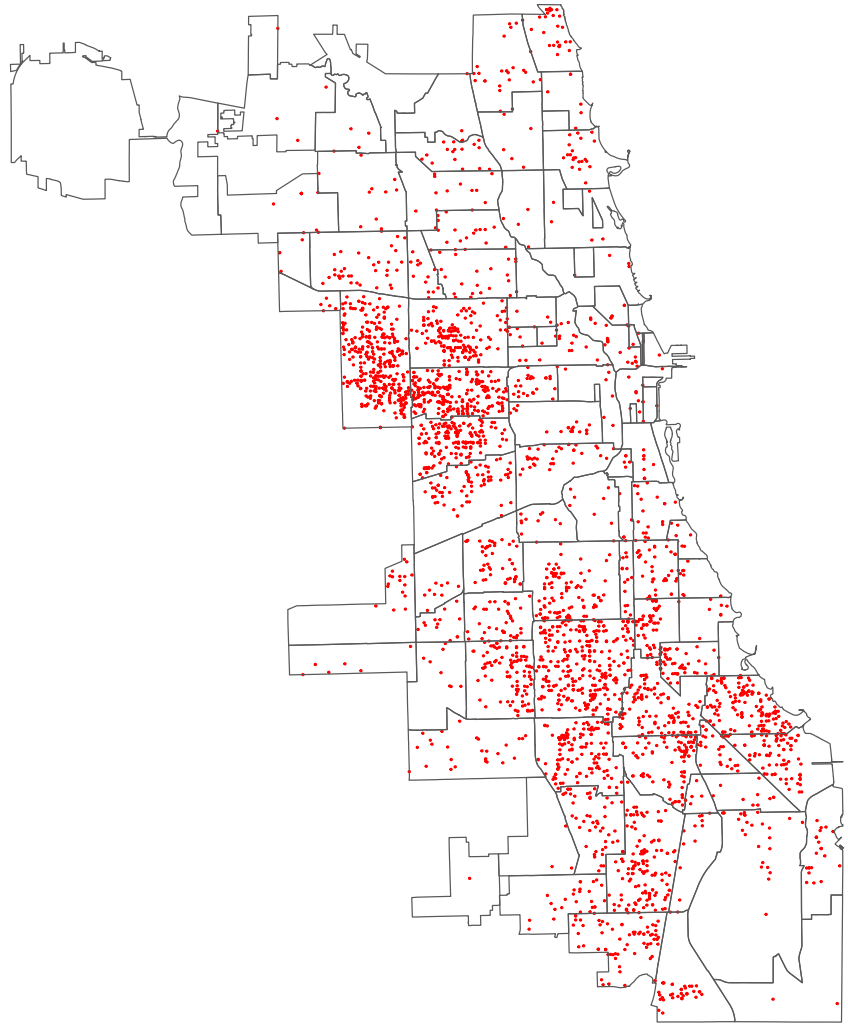
Read in the data

```
crime_data <- read.csv("../data/2016-2020-chicago-crime.csv")

neighborhoods <- st_read("../data/Boundaries - Neighborhoods.geojson",quiet=T)

census_tracts <- st_read("../data/Boundaries - Census Tracts - 2010.geojson", q

homicide <- crime_data %>%
  filter(Primary.Type=="HOMICIDE")
```

Code: Homicides in Chicago Neighborhoods

```
neighborhoods_map <- neighborhoods %>%
  ggplot() +
  geom_point(data = homicide, aes(x=Longitude, y=Latitude), shape=1, color="red
  geom_sf(aes(geometry=geometry), fill="transparent", linewidth=.25) +
  theme(axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks = element_blank(),
        rect = element_blank(),
        legend.position = "bottom") +
  labs(x="", y="")#, title = "Homicides in Chicago (2016-2019)",
      # subtitle = "By Neighborhood")
```

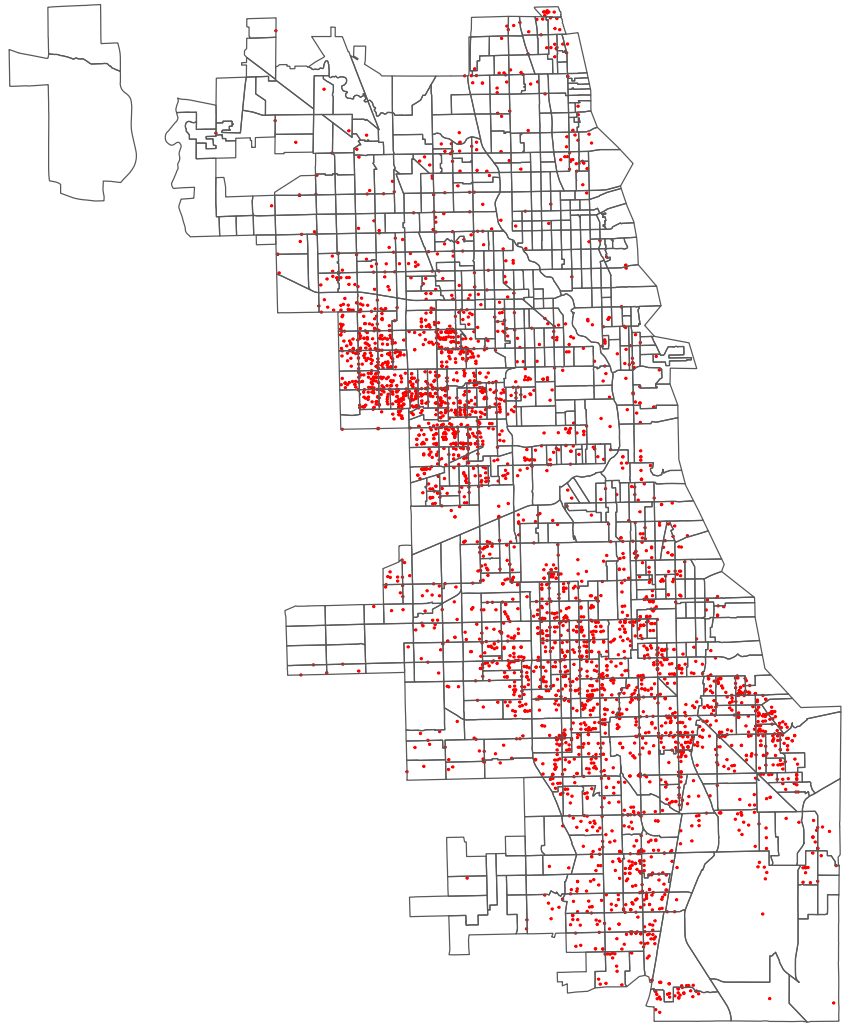
Output: Homicides in Chicago (2016-2019) by Neighborhood



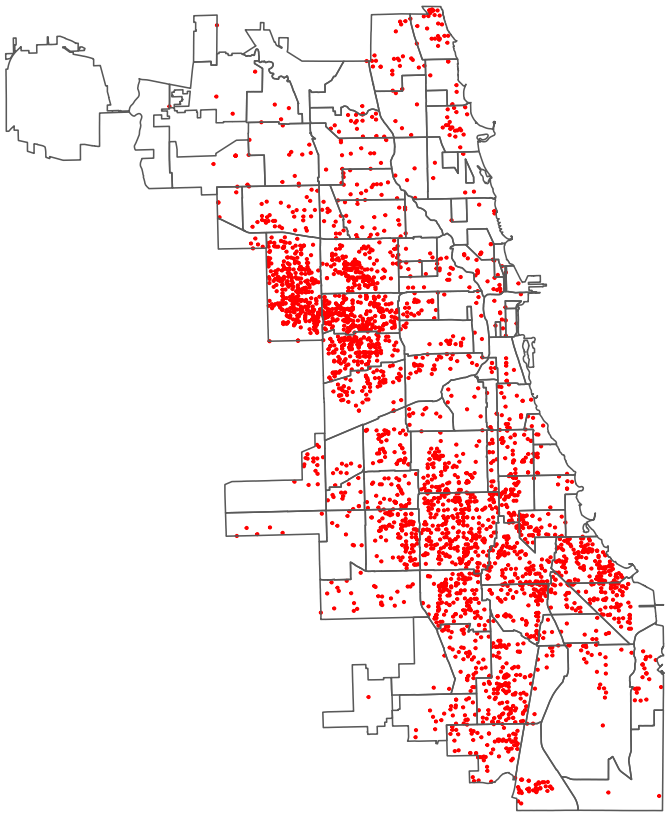
Code: Homicides in Chicago Census Tracts

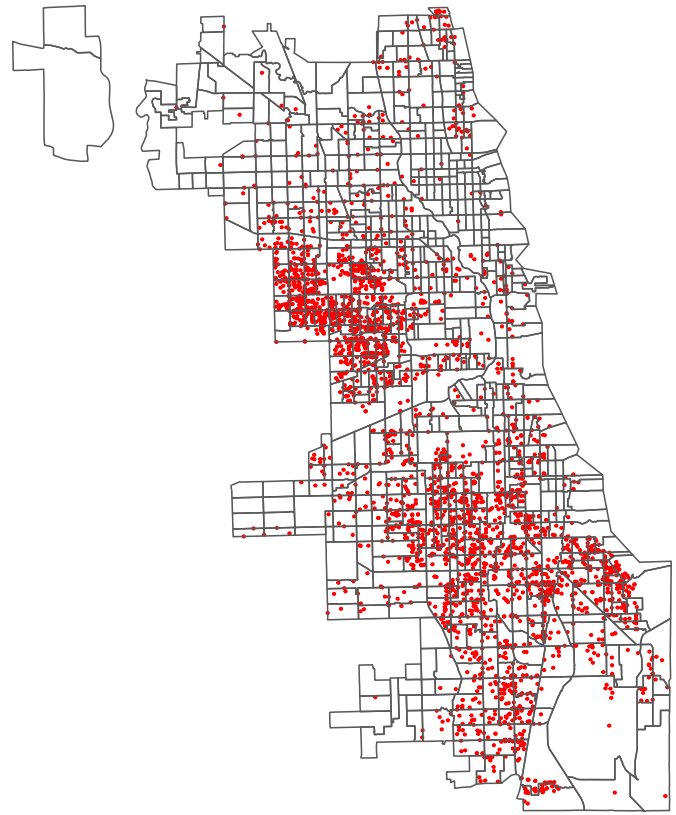
```
tracts_map <- census_tracts %>%
  ggplot() +
  geom_point(data = homicide, aes(x=Longitude, y=Latitude), shape=1, color="red
  geom_sf(aes(geometry=geometry), fill="transparent", linewidth=.25) +
  theme(axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks = element_blank(),
        rect = element_blank(),
        legend.position = "bottom") +
  labs(x="", y="") #, title = "Homicides in Chicago (2016-2019)",
      # subtitle = "By Neighborhood")
```

Output: Homicides in Chicago (2016-2019) by Census Tract



Comparison





Let's make neighborhood and census tract heat maps now

```
point_sf <- st_as_sf(homicide, coords = c("Longitude", "Lat

#crs=4326 WGS 1984 latitudes and longitudes (EPSG 4326), t

neighborhoods$homicides <- lengths(st_intersects(neighborho
census_tracts$homicides <- lengths(st_intersects(census_tra
```

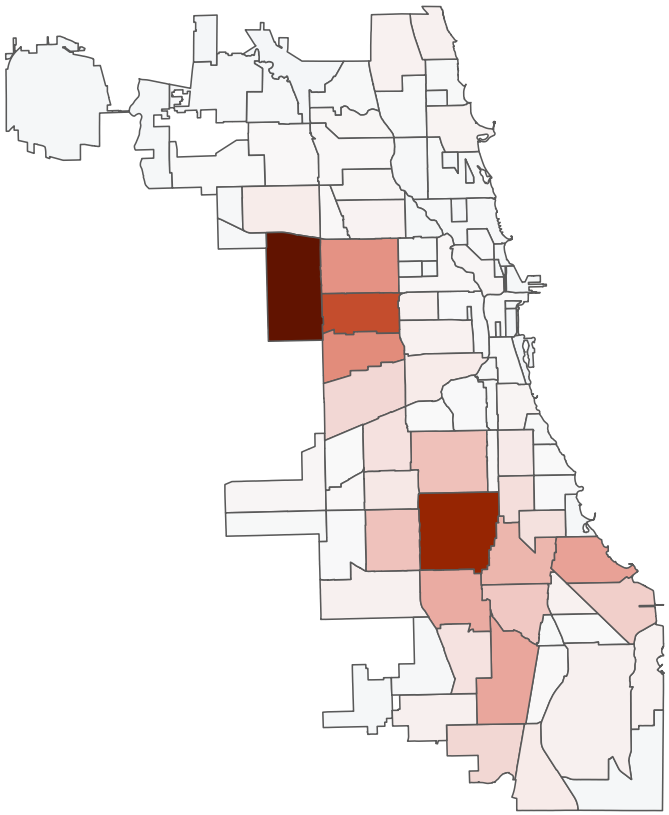
Neighborhood Heat Map

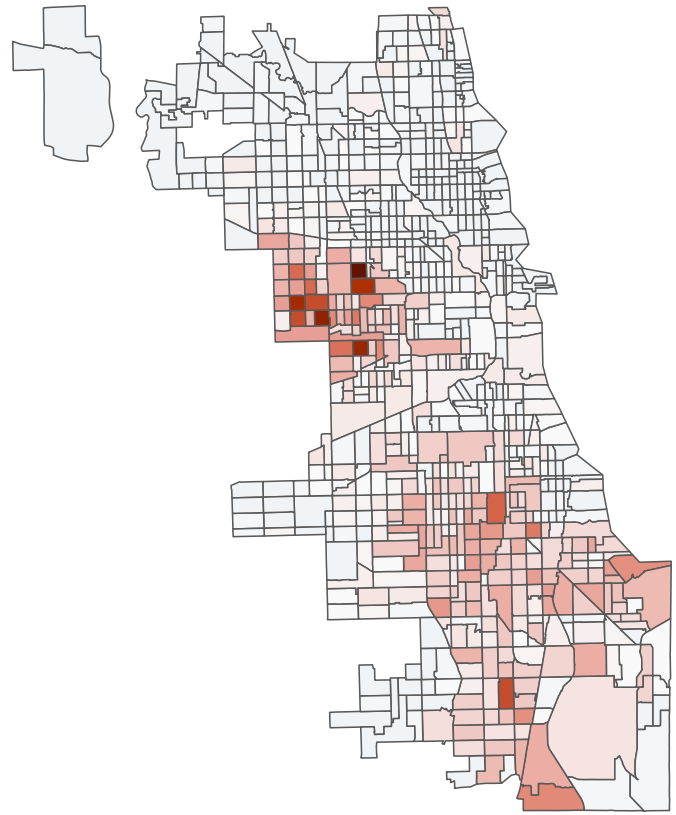
```
neighborhood_heat_map <- neighborhoods %>%
  ggplot() +
  geom_sf(aes(geometry=geometry, fill=homicides), linewidth
  theme(axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks = element_blank(),
        rect = element_blank(),
        legend.position = "none") +
  labs(x="", y="") +
  scale_fill_continuous_divergingx(palette = 'RdBu', mid =
```

Census Tract Heat Map

```
tract_heat_map <- census_tracts %>%
  ggplot() +
  geom_sf(aes(geometry=geometry, fill=homicides), linewidth
  theme(axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks = element_blank(),
        rect = element_blank(),
        legend.position = "none") +
  labs(x="", y="") +
  scale_fill_continuous_divergingx(palette = 'RdBu', mid =
```

Heat Map Comparison





Point Pattern Analysis

- Heat maps and other visuals can be informative but often want to do more rigorous analysis of spatial point patterns
- We are interested in understanding the spatial distribution of homicides in Chicago:
  - Are homicide events distributed randomly, clustered, or dispersed?
  - What is the overall density or intensity of events across the study area? What are the underlying processes driving the spatial distribution of points?
  - Are there statistically significant clusters of events in the study area?
  - Where are the areas with significantly higher or lower homicide event densities compared to the surrounding areas?
- We can apply these questions and framework to analyzing any point pattern data within a region

Kernel Density Estimation

Spatial Point Patterns

- Countable sets of points that arise as realizations of stochastic point processes taking values in A $\subset \mathbb{R}^2$
- A spatial point pattern can be denoted as $\{x_1, x_2, \ldots, x_{N(A)}\}$, where $N(A)$ is the number of points in A. Note that $N(A)$ is a random variable.

```
library(spatstat)

borders = st_transform(neighborhoods$geometry, crs=6454)
#Point patterns must be projected (rather than being lat/l

ppp.homicide = as.ppp(st_transform(point_sf$geometry, crs=6

density.homicide = density.ppp(ppp.homicide, w=as.owin(bord
density.homicide = density.ppp(ppp.homicide, sigma=500)



plot(density.homicide, col=colorRampPalette(c("navy", "gray
plot(borders, col=NA, border="gray", add=T)
```

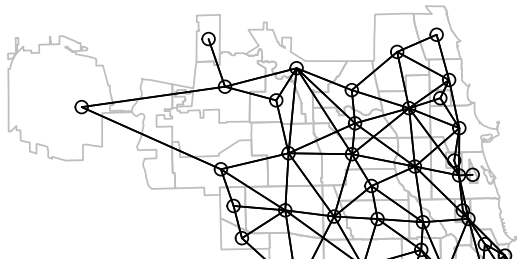
Global and Local Moran's I

Getis-Ord / G*

Wrap-up

Wrap-up

```
# talk about different weight matrices options
# contiguous, inverse etc., why use one over the other
# talk about how these are used, why people use in dependen
library(spdep)

neighbors = poly2nb(as_Spatial(neighborhoods), queen=T)

weights = nb2listw(neighbors, style="W", zero.policy=T)

plot(st_geometry(neighborhoods), border="gray")

plot(neighbors, coords=st_coordinates(st_centroid(neighborh
```



Next steps:

- descriptions/slides for each type of analysis
- consider using different geographical boundaries, ie census tracts, highlight modifiable area unit problem + parameters for different methods
- resources on spatial regression, future directions, etc.
- bigger data/computation: plug for Eric's workshop!!!

package dependencies

some other demographic/neighborhood characteristics that are associated with higher crimes, etc.

add in population data for the neighborhoods and census tracts

if you have other high frequency data, ie cell phone data or all crime data as benchmark for computation time spatial heterogeneity ??

source: https://michaelminn.net/tutorials/r-crime/