Introduction to GIS and Disease Mapping with R

Esteban Correa, PhD

Agenda

- Brief history and Justification
- Basics of spatial data science in R
- Coordinate reference system
- Disease mapping example

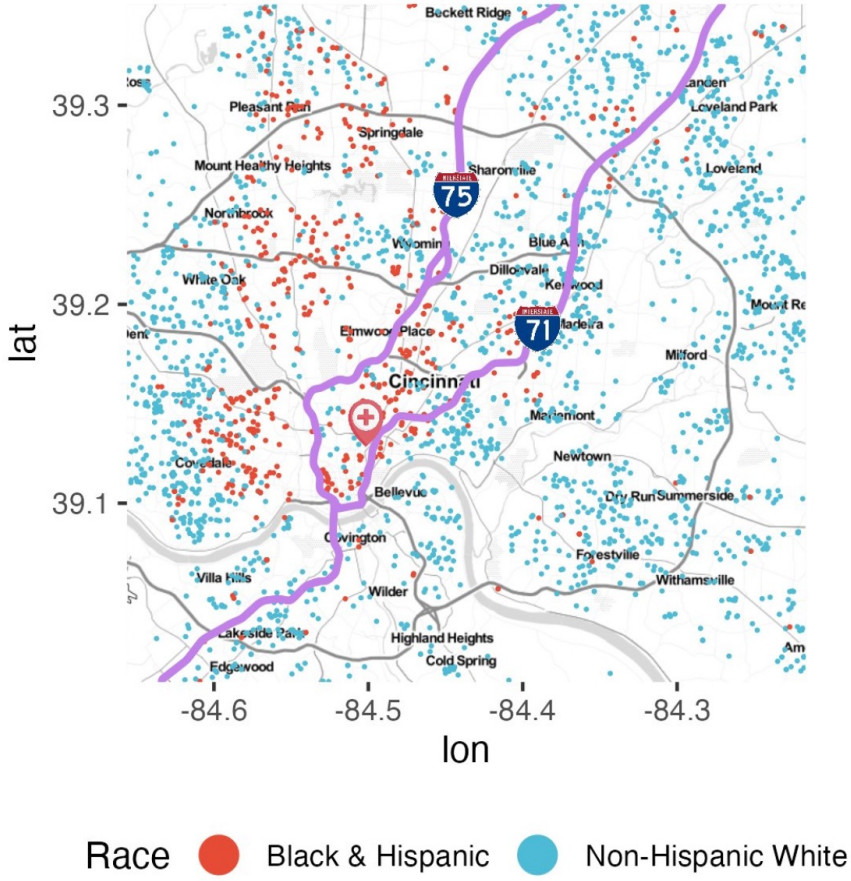
History

- Leonard Ludwig Finke (1747-1837) produced the first world map of diseases in 1789.
- It was based on a zoological map, but Finke used the names of indigenous diseases.
- Finke used a latitudinal framework due to correlation with climate and environmental regions.

[Taken from Barret, 2000]

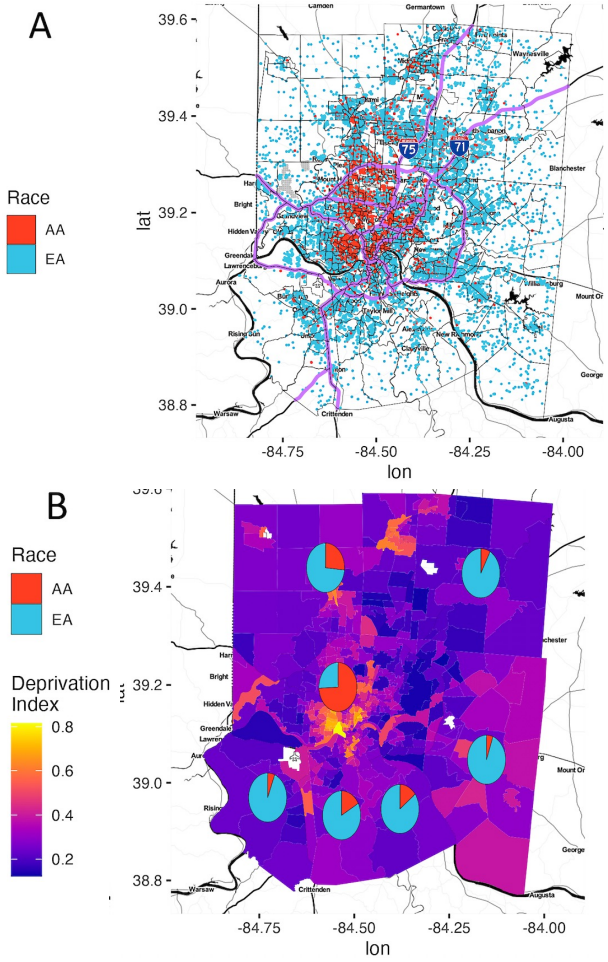
Maps or no maps

- Helps to reveal spatial disparities.
- Allows to incorporate explicit landscape elements and exposures.
- Well suited for targeted interventions based on risk areas.



Maps or no maps

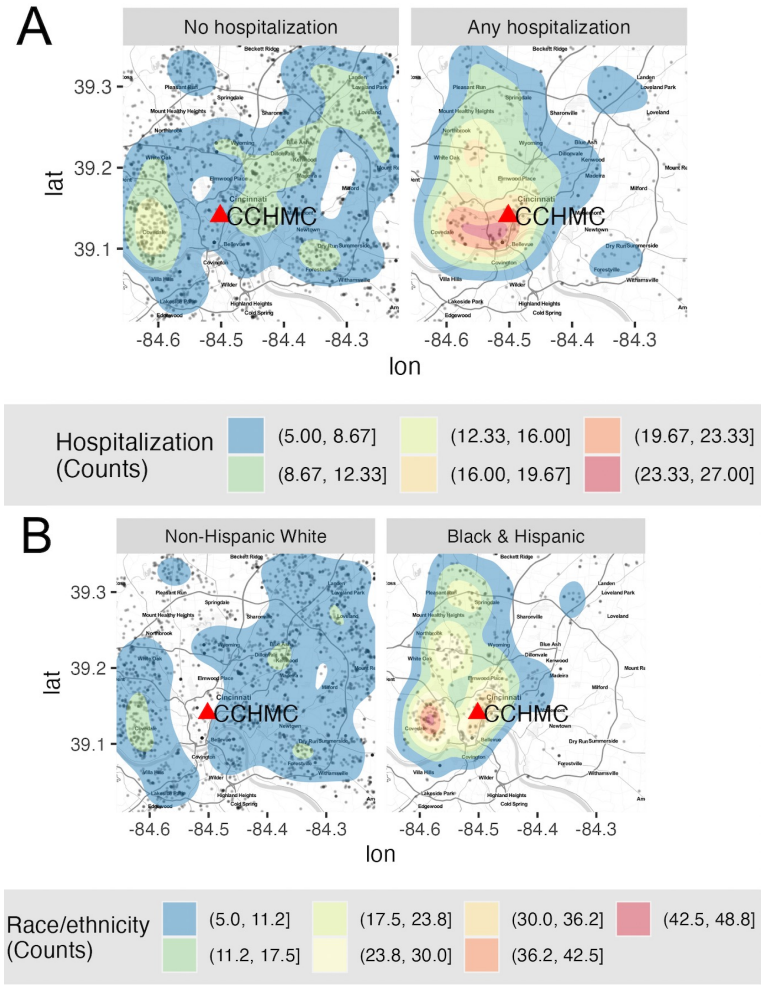
- Helps to reveal spatial disparities.
- Allows to incorporate explicit landscape elements and exposures.
- Well suited for targeted interventions based on risk areas.



[Taken from Correa et al, https://doi.org/10.1016/j.jaci.2022.07.024]

Maps or no maps

- Helps to reveal spatial disparities.
- Allows to incorporate explicit landscape elements and exposures.
- Well suited for targeted interventions based on risk areas.



[Taken from Correa et al, 2022]

Spatial data

GIS view

Geometry sets with attributes in tables.

Real view

Plain text, .csv files requiring some geo-processing.

Data science view

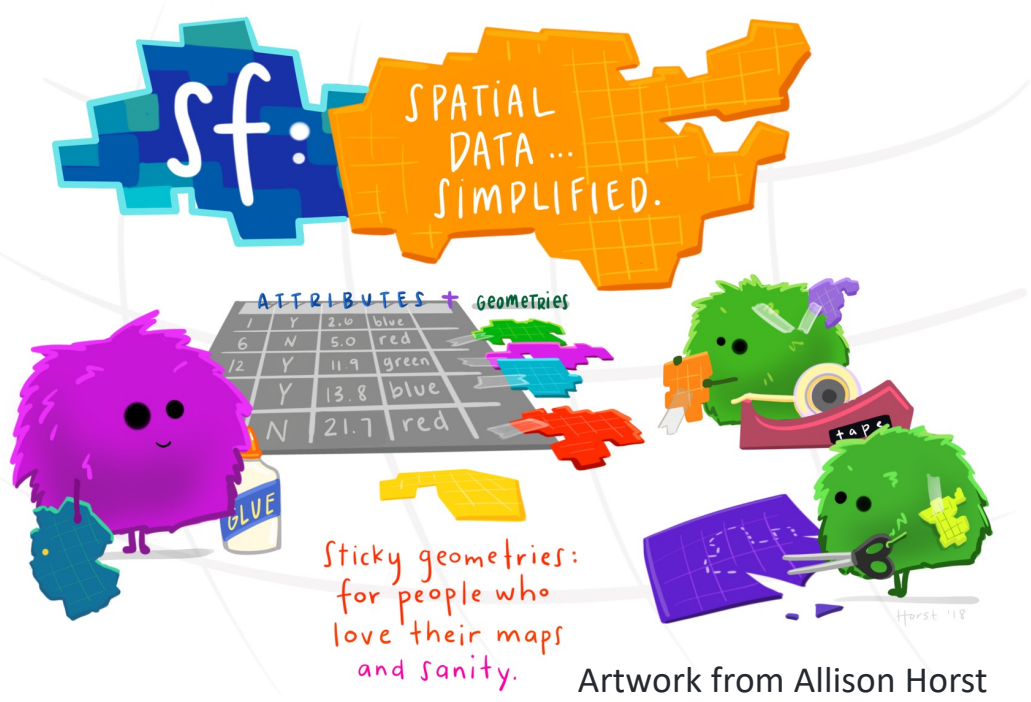
Tables extended with geometries and coordinate system.

Spatial Data: real view

RAW location information

	DATE_REPORTED	OFFENSE	ADDRESS_X	LONGITUDE_X	LATITUDE_X
1	06/01/2022 10:44:00 AM	THEFT	28XX CENTRAL PKWY	-84.53237	39.13435
2	06/01/2022 10:14:16 AM	CRIMINAL DAMAGING/ENDANGERING	3XX W COURT ST	-84.52035	39.10579
3	06/01/2022 10:00:00 AM	BREAKING AND ENTERING	21XX MADISON RD	-84.45838	39.13338
4	06/01/2022 10:00:00 AM	BREAKING AND ENTERING	21XX MADISON RD	-84.45960	39.13460
5	06/01/2022 09:48:41 AM	MISUSE OF CREDIT CARD	XX E UNIVERSITY AV	-84.50745	39.13301
6	06/01/2022 09:46:15 AM	THEFT	34XX PRICE AV	-84.56776	39.10664
7	06/01/2022 09:38:01 AM	THEFT	19XX STATE AV	-84.54381	39.12203
8	06/01/2022 09:13:00 AM	THEFT	5XX ELBERON AV	-84.56642	39.10127
9	06/01/2022 08:45:00 AM	MENACING	30XX COLERAIN AV	-84.53776	39.13779
10	06/01/2022 08:45:00 AM	TELEPHONE HARASSMENT	30XX COLERAIN AV	-84.53867	39.13968

Spatial Data: DS view



Artwork from Allison Horst

Attributes can be:

- Area of a census tract
- Population of a county
- Name of street

Geometries can be:

- Polygons
- Points
- Line, Multiline

Spatial Data: Polygons

```r
library(tigris)
options(tigris_use_cache = TRUE)

countiesOH<-counties(state = "OH") %>%
  dplyr::select(GEOID,NAMELSAD,ALAND)
```

```
Retrieving data for the year 2020
Using FIPS code '39' for state 'OH'
```

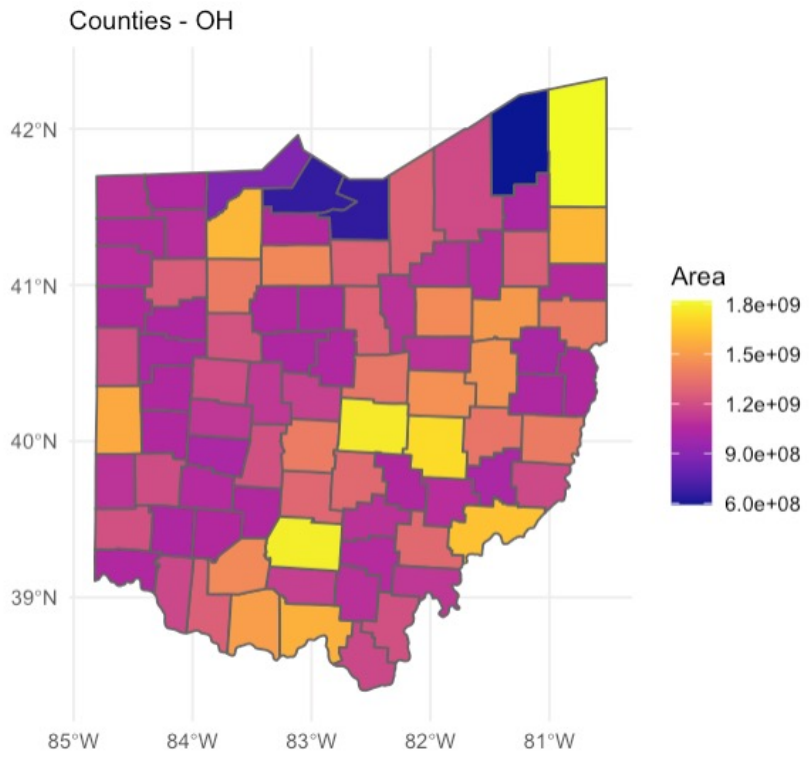
Attributes

GEOID	NAMELSAD	ALAND	geometry
39001	Adams County	1512221266	*MULTIPOLYGON (((-83.68696 3...*
39003	Allen County	1042587389	*MULTIPOLYGON (((-84.39719 4...*
39005	Ashland County	1095530837	*MULTIPOLYGON (((-82.4179 40...*
39007	Ashtabula County	1818345034	*MULTIPOLYGON (((-81.00363 4...*
39009	Athens County	1304431980	*MULTIPOLYGON (((-82.26024 3...*
39011	Auglaize County	1039590216	*MULTIPOLYGON (((-84.45562 4...*
39013	Belmont County	1378201570	*MULTIPOLYGON (((-81.23066 4...*
39015	Brown County	1267911780	*MULTIPOLYGON (((-84.03663 3...*
39017	Butler County	1208290720	*MULTIPOLYGON (((-84.59153 3...*
39019	Carroll County	1022040910	*MULTIPOLYGON (((-81.26474 4...*

Geometries

Spatial Data: Polygons

```
ggplot()+
  geom_sf(data=countiesOH,aes(fill=ALAND))+
  scale_fill_viridis(option = "C")+
  labs(subtitle = "Counties - OH",fill="Area")+
  theme_minimal()
```



Spatial Data: points

```r
# Converting data to spatial-friendly format
crimesRV <-
  read.csv("data/PDI__Police_Data_Initiative__Crime_Incidents.csv")
crimesRV1 <- crimesRV %>%
  mutate(
    DATE_REPORTED2 = as.Date(crimes$DATE_REPORTED, format =  "%m/%d/%Y"),
    year = year(DATE_REPORTED2)
  ) %>%
  filter(DATE_REPORTED2 >= mdy("01-01-2009")) %>%
  drop_na(c("LONGITUDE_X", "LATITUDE_X", "year")) %>%
  dplyr::select(c(
    "year",
    "DATE_REPORTED2",
    "OFFENSE",
    "ADDRESS_X",
    "LONGITUDE_X",
    "LATITUDE_X"
  ))

crimesDSview = crimesRV1 %>%
  dplyr::filter(year %in% c(2019)) %>%  # filterv2019 crimes
  st_as_sf(coords = c("LONGITUDE_X", "LATITUDE_X")) %>% # set coordinates
  st_set_crs("EPSG:4326") # set geographic CRS
```

DATE_REPORTED	OFFENSE	ADDRESS_X	geometry
12/31/2019 11:53:00 PM	DOMESTIC VIOLENCE	19XX WESTMONT LN	POINT (-84.57331 39.1236)
12/31/2019 11:52:00 PM	IMPROPERLY DISCHARGING FIREARM AT/INTO HABITA...	7XX CHESTNUT ST	POINT (-84.52755 39.10692)
12/31/2019 11:50:00 PM	THEFT	55XX KIRBY AV	POINT (-84.56813 39.19412)
12/31/2019 11:45:00 PM	CRIMINAL DAMAGING/ENDANGERING	47XX GLENWAY AV	POINT (-84.59687 39.11617)
12/31/2019 11:10:00 PM	DOMESTIC VIOLENCE	26XX MONTANA AV	POINT (-84.58488 39.15131)
12/31/2019 10:50:00 PM	DOMESTIC VIOLENCE	23XX HARRISON AV	POINT (-84.57633 39.13721)
12/31/2019 10:10:00 PM	FELONIOUS ASSAULT	16XX DEWEY AV	POINT (-84.58244 39.11949)
12/31/2019 10:00:00 PM	DOMESTIC VIOLENCE	23XX MAPLEWOOD AV	POINT (-84.50496 39.12277)
12/31/2019 09:59:53 PM	THEFT	17XX VINE ST	POINT (-84.51539 39.11514)
12/31/2019 09:57:00 PM	THEFT	6XX DERBY AV	POINT (-84.51504 39.17108)

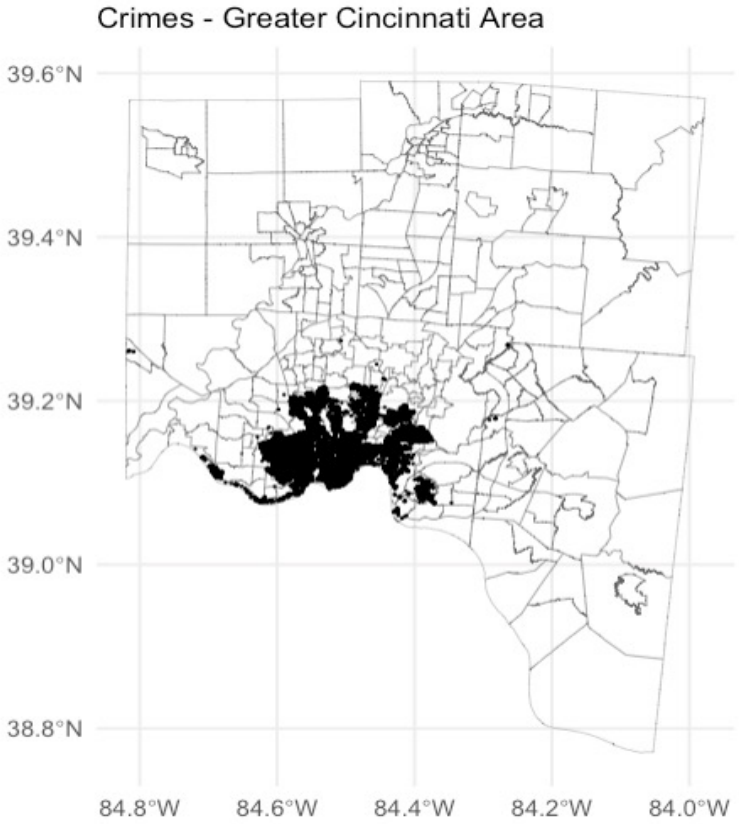
23XX HARRISON AV

Spatial Data: points

```
coi.list<-c(
  "39017",
  "39165",
  "39061",
  "39025"
  )

tractsGC<-tracts(state = "OH",year = 2018)%>%
  dplyr::mutate(county=as.factor(substr(GEOID, 1, 5)))%>%
  dplyr::filter(county %in% coi.list)

ggplot ()+
  geom_sf(data=tractsGC,fill=NA,size=0.1)+
  geom_sf(data=crimesDSview,size=0.01)+
  labs(subtitle = "Crimes - Greater Cincinnati Area")+
  theme_minimal()
```

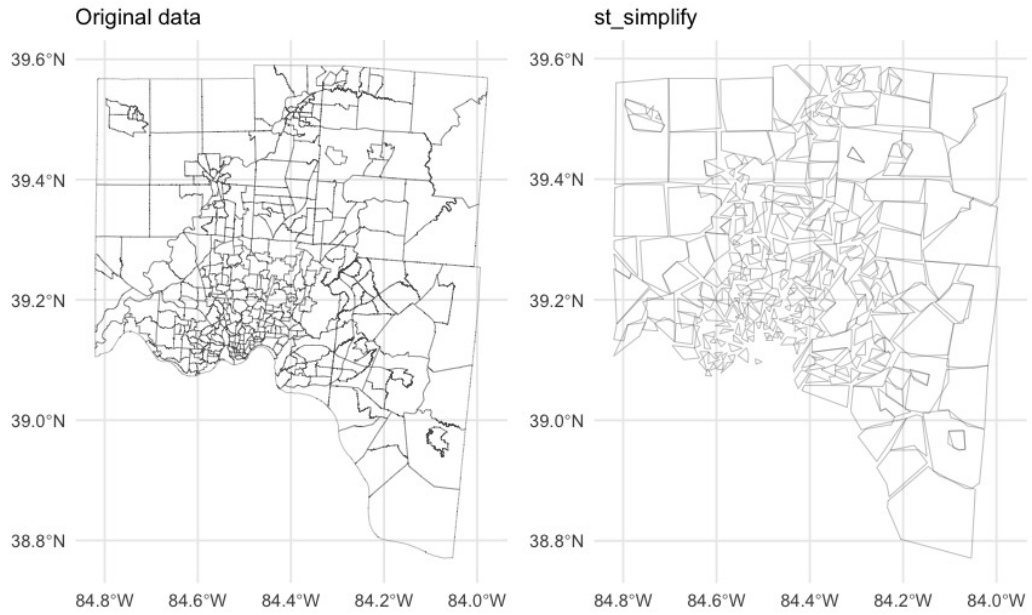


Basic operations

Simplification

```r
tractsGC_simpl<- st_simplify(tractsGC, dTolerance = 1000)  # 1000 m
object.size(tractsGC)
object.size(tractsGC_simpl)
```

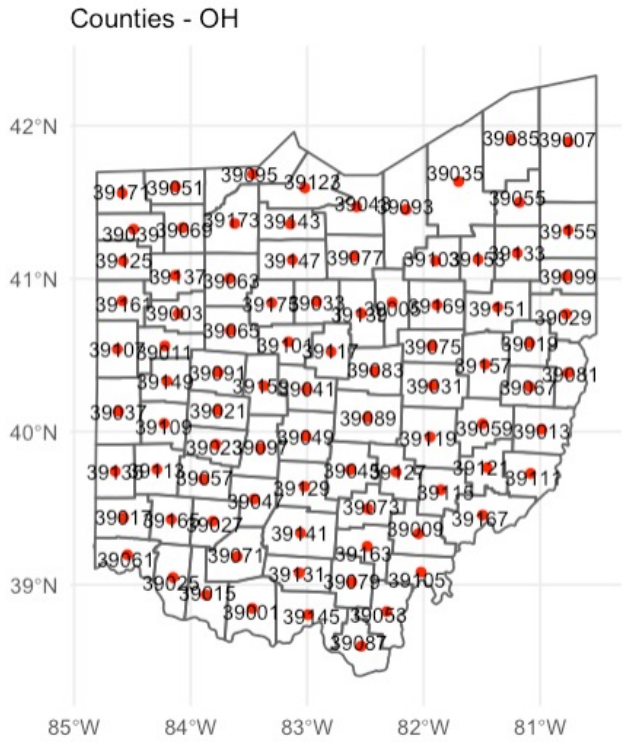
```
2101504 bytes
455696 bytes
```



Centroids

```r
countiesOH2 <- cbind(countiesOH, st_coordinates(st_centroid(countiesOH)))
centroids <- st_centroid(countiesOH)

ggplot()+
  geom_sf(data=countiesOH2,fill=NA)+
  geom_sf(data=centroids, color="red")+ #centroid at each county
  geom_sf_text(data=countiesOH2,aes(label=GEOID),size=3)+ # label for each polygon
  labs(subtitle = "Counties - OH",fill="Area")+
  theme_minimal()
```



Spatial Join

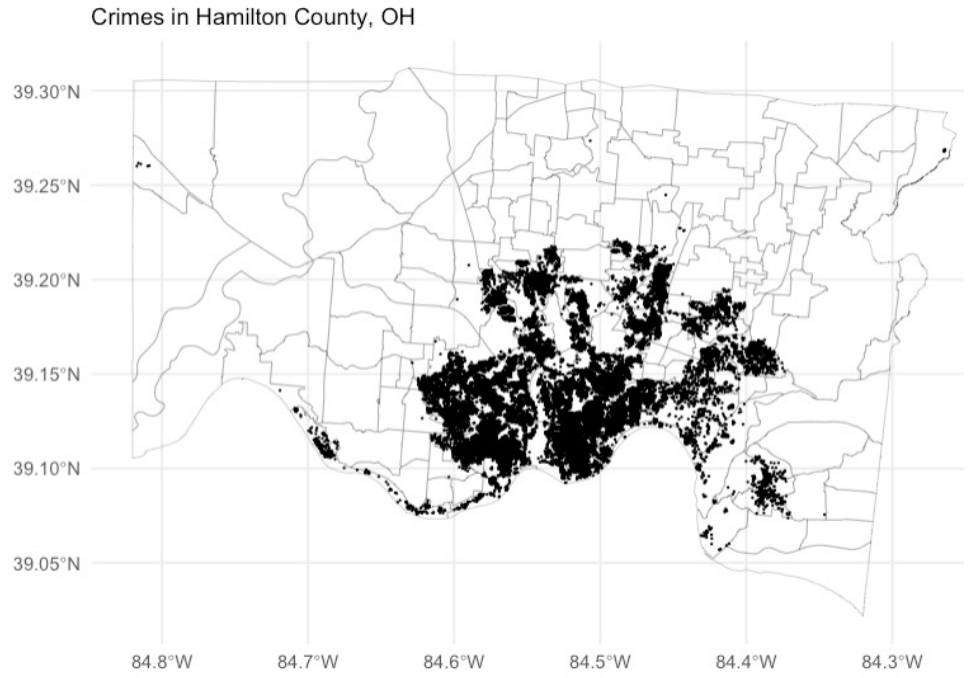
```r
#Filter hamilton tracts
tractsHamilton<-tractsGC %>%
  dplyr::filter(grepl("39061",GEOID))

ggplot ()+
  geom_sf(data=tractsHamilton,fill=NA,size=0.1)+
  labs(subtitle = "Original data")+
  theme_minimal()

#Set same coordinate system
tractsHamilton
st_crs(tractsHamilton)<-st_crs(crimesDSview)
crimesDSview

#Spatial Join
crimesHamilton<-st_join(crimesDSview,tractsHamilton) %>%
  filter(!is.na(GEOID))

ggplot ()+
  geom_sf(data=tractsHamilton,fill=NA,size=0.1)+
  geom_sf(data=crimesHamilton,size=0.01)+
  labs(subtitle = "Crimes in Hamilton County, OH")+
  theme_minimal()
```



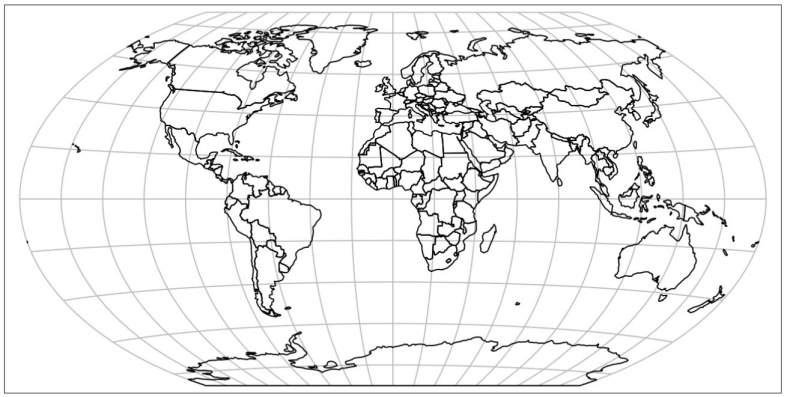
Coordinate reference system

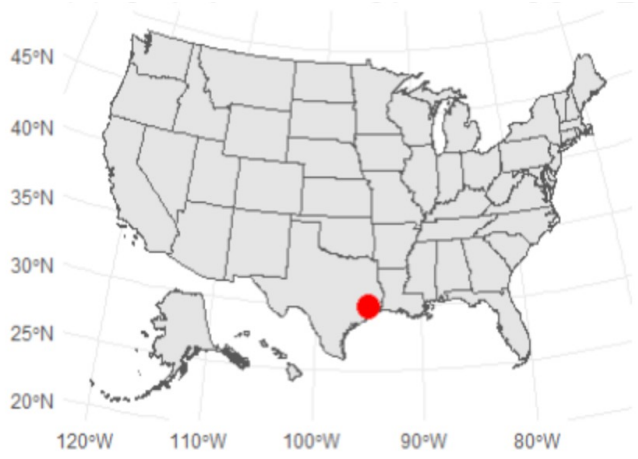
When to project?

- When performing geometric operations:
  - Measure of distance
  - Buffers
- Visualization-related stuff

Tips?

- There exist no all-purpose projections.
- Always check spatial objects are in the same coordinate system



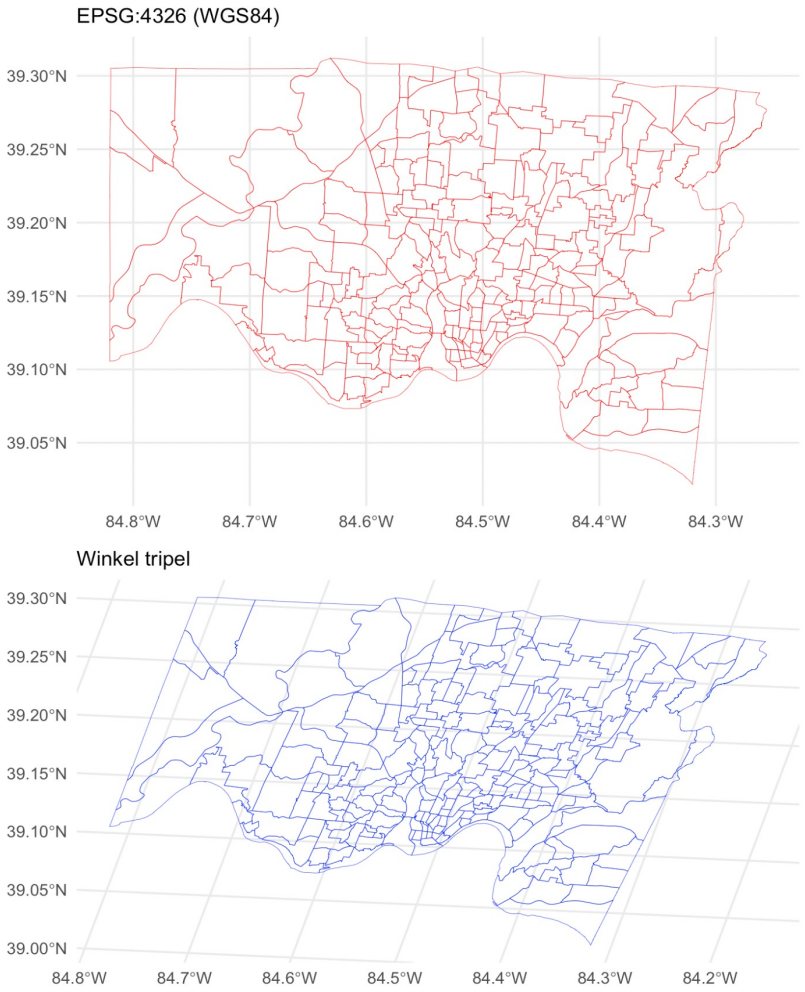


Reprojection

```
# convert a dataset to a spatial object
crimesDSview = crimesRV1 %>%
  dplyr::filter(year %in% c(2019)) %>%  # filter 2019 crimes
  st_as_sf(coords = c("LONGITUDE_X",
                      "LATITUDE_X")) %>% # set coordinates
  st_set_crs("EPSG:4326") # set geographic CRS
```

```
#Set same coordinate system
st_crs(tractsHamilton)<-st_crs(crimesDSview)
```

```
# reproject
tractsHamiltonWintri = st_transform(tractsHamilton,
                                    crs = "+proj=wintri")
```



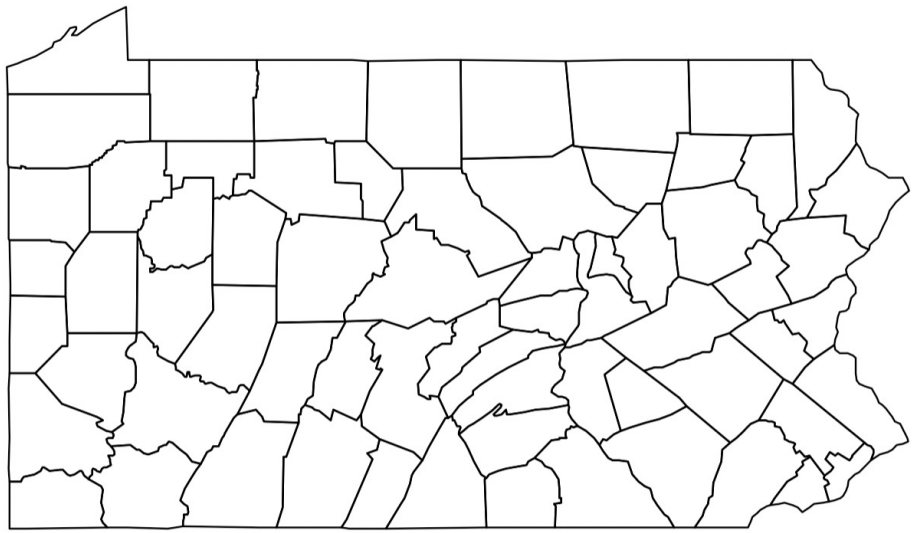
Disease mapping example

R package: SpatialEpi

Dataset: pennLC

Standardized incidence ratio (SIR)

$$\mathrm{SIR}_i = Y_i / E_i.$$



[Taken from https://www.paulamoraga.com/book-geospatial/]

Disease mapping example

```
library(SpatialEpi)

pennLC
map <- pennLC$spatial.polygon # GIS view
plot(map)
```

```
# Summarize by each county
d <- group_by(pennLC$data, county) %>%
  summarize(Y = sum(cases))

# We order by each strata to hold original
# population distribution
pennLC$data <- pennLC$data[order(
  pennLC$data$county,
  pennLC$data$race,
  pennLC$data$gender,
  pennLC$data$age
), ]

E <- expected(
  population = pennLC$data$population,
  cases = pennLC$data$cases, n.strata = 16
)
```

county <fctr>	Y <int>
adams	55
allegheny	1275
armstrong	49
beaver	172
bedford	37
berks	308
blair	127
bradford	59
bucks	454
butler	158

Disease mapping example

```
d$E <- E[match(d$county, unique(pennLC$data$county))]
head(d)
d$SIR <- d$Y / d$E

mapDS <- st_as_sf(map)
mapDS$county=names(map)
mapDS2<-mapDS %>%
  left_join(d,by="county")

highSIR<-mapDS2 %>%
  filter(SIR>1.1)
```

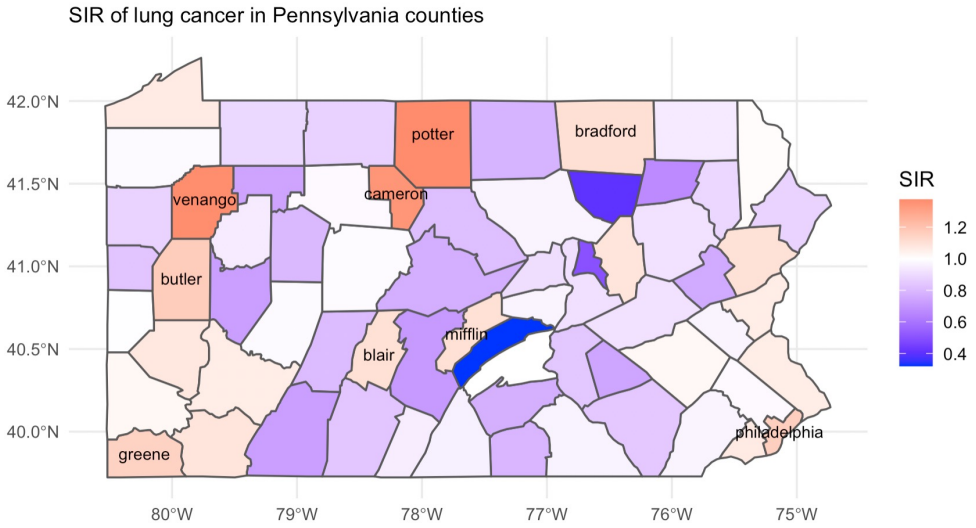
county	Y	E	SIR	geometry
adams	55	69.627305	0.7899200	*POLYGON ((-77.4467 39.96954...*
allegheny	1275	1182.428036	1.0782897	*POLYGON ((-80.14534 40.6742...*
armstrong	49	67.610123	0.7247435	*POLYGON ((-79.21142 40.9091...*
beaver	172	172.558055	0.9967660	*POLYGON ((-80.1568 40.85189...*
bedford	37	44.190132	0.8372910	*POLYGON ((-78.38063 39.7288...*
berks	308	300.705979	1.0242563	*POLYGON ((-75.53303 40.4508...*
blair	127	115.069655	1.1036793	*POLYGON ((-78.11707 40.7373...*
bradford	59	53.237644	1.1082384	*POLYGON ((-76.14609 42.0035...*
bucks	454	428.797481	1.0587749	*POLYGON ((-74.97153 40.0554...*
butler	158	134.797705	1.1721268	*POLYGON ((-80.14534 40.6742...*

county	Y	E	SIR	geometry
blair	127	115.069655	1.103679	*POLYGON ((-78.11707 40.7373...*
bradford	59	53.237644	1.108238	*POLYGON ((-76.14609 42.0035...*
butler	158	134.797705	1.172127	*POLYGON ((-80.14534 40.6742...*
cameron	8	5.945905	1.345464	*POLYGON ((-78.10561 41.2185...*
greene	38	33.093112	1.148275	*POLYGON ((-80.4089 39.72316...*
mifflin	45	40.774630	1.103627	*POLYGON ((-77.36649 40.8404...*
philadelphia	1415	1219.102696	1.160690	*POLYGON ((-74.97153 40.0554...*
potter	22	16.003210	1.374724	*POLYGON ((-77.76183 42.0035...*
venango	70	51.141014	1.368764	*POLYGON ((-79.47498 41.3732...*

Disease mapping example

```
highSIR2 <- cbind(highSIR,
                  st_coordinates(st_centroid(highSIR)))

ggplot()+
  geom_sf(data=mapDS2,aes(fill = SIR))+
  scale_fill_gradient2(
    midpoint = 1,
    low = "blue",
    mid = "white",
    high = "red"
  ) +
  coord_sf(crs="EPSG:4326")+
  geom_sf_text(data=highSIR,
               aes(label=county),
               size=3)+ # label for each polygon
  labs(subtitle="SIR of lung cancer in Pennsylvania counties")+
  theme_minimal()
```



More resources

Geocoding services

- Google Maps API
- degauss.org

Remote sensing resources

- google earth engine
- NASA

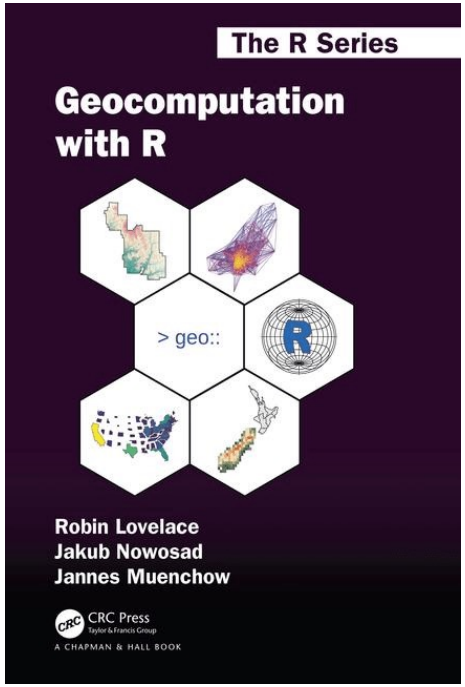
Spatial indexing

- h3

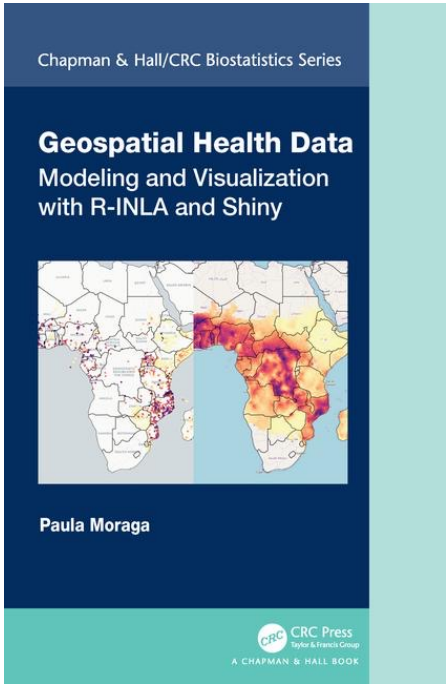
Rspatial packages

- tidyverse (dplyr, ggplot2) or tmap + base R
- sf or sp (gis view)
- terra
- raster/rgdal

Books



https://geocompr.robinlovelace.net/



https://www.paulamoraga.com/book-geospatial/

Thanks

@maurosc3ner

/maurosc3ner