Spatial data analysis (1)

Yue Jiang

Duke University

A disclaimer

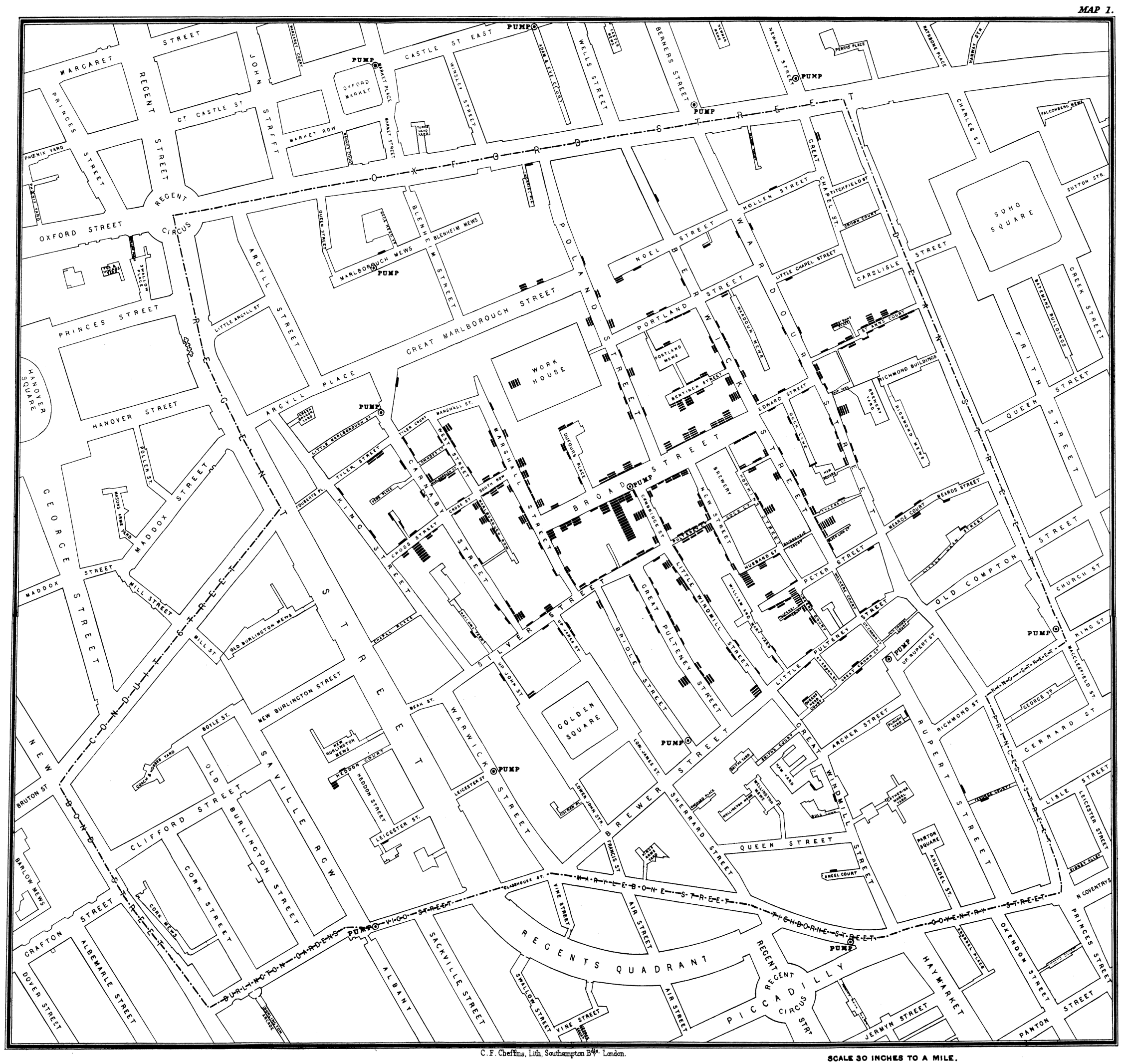
The following material was used during a live lecture. Without the accompanying oral comments and discussion, the text is incomplete as a record of the presentation. A full recording may be found via Zoom on the course Sakai site.

Spatial data

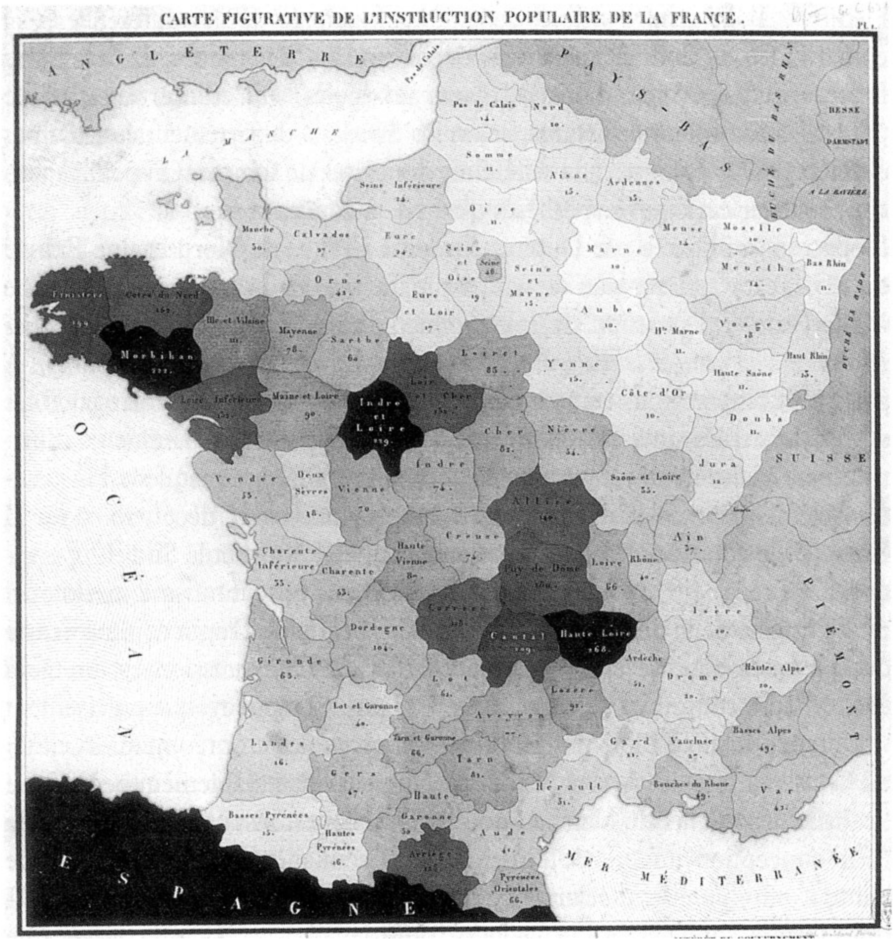
Spatial data are an important class of data. Today, we will focus on exploratory data analysis, understanding spatial relationships, and detecting patterns and trends.

Analysis of spatial data should reflect spatial structure!

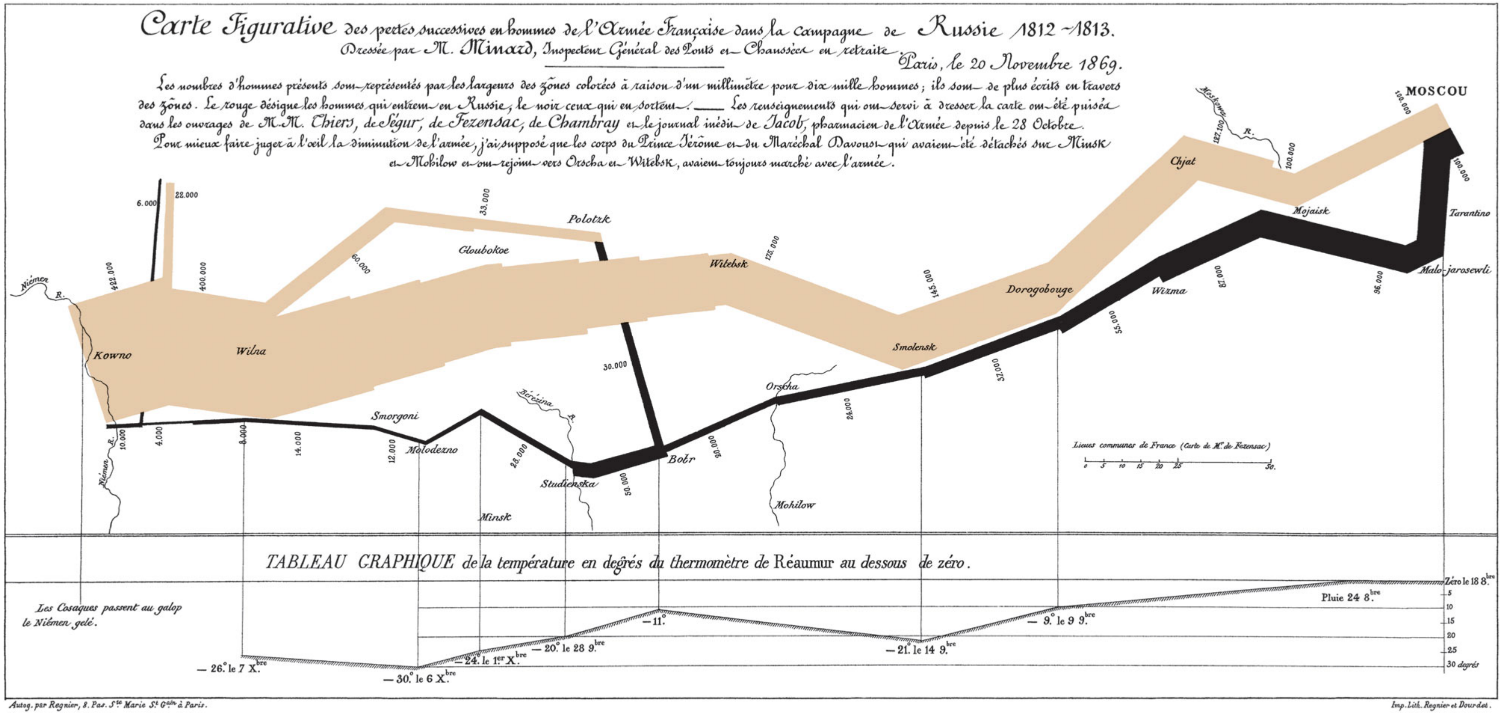
1854 London cholera outbreak



1826 French literacy map



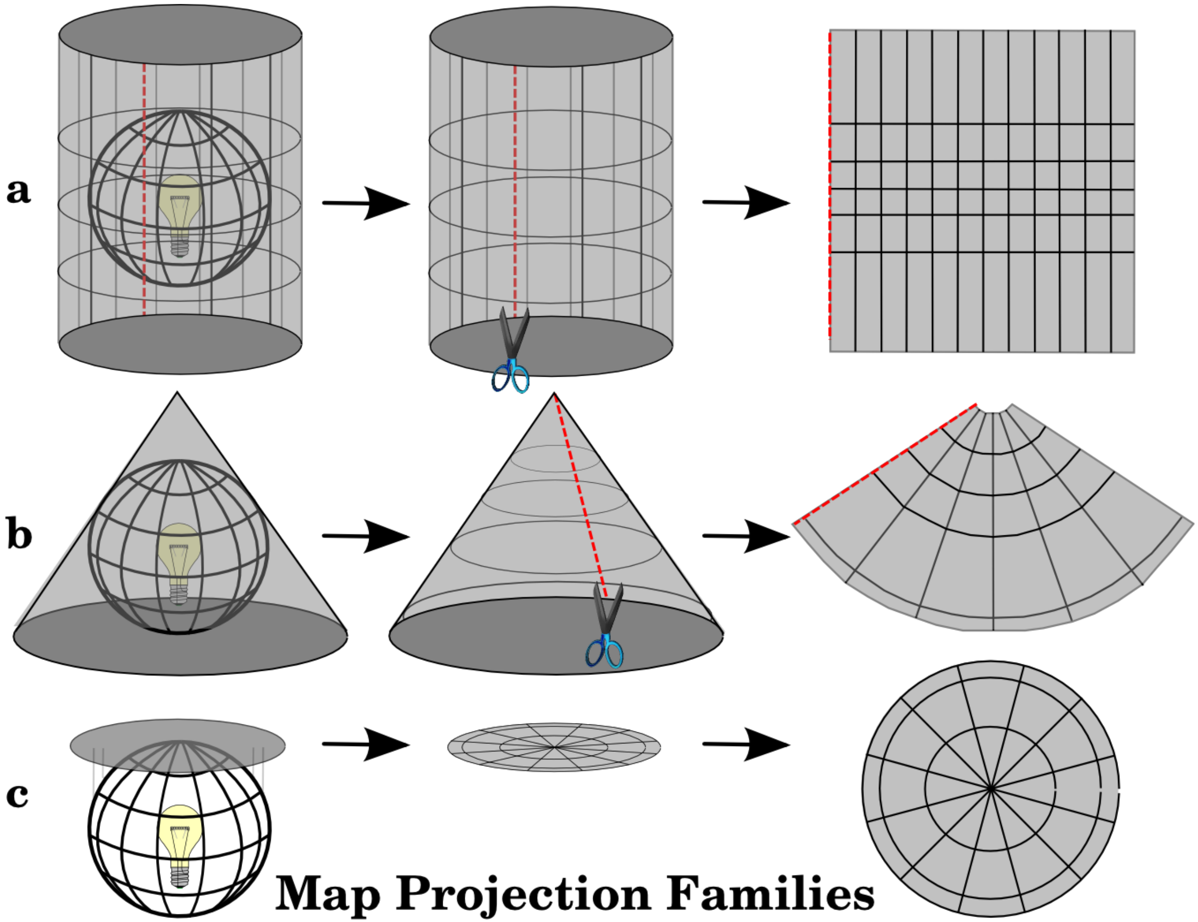
Napoleon's 1812 Russia Campaign



Many others!

- Migrations
- World Population Density
- Global Power

Spatial data are different



Graphic from QGIS documentation.

Spatial data are different

A simple feature is a standard way to describe how real-world spatial objects (country, building, tree, road, etc) can be represented by a computer.

The package `sf` implements simple features and other spatial functionality using **tidy** principles for *vector* graphics.

Simple features have a geometry type. Common choices are below.

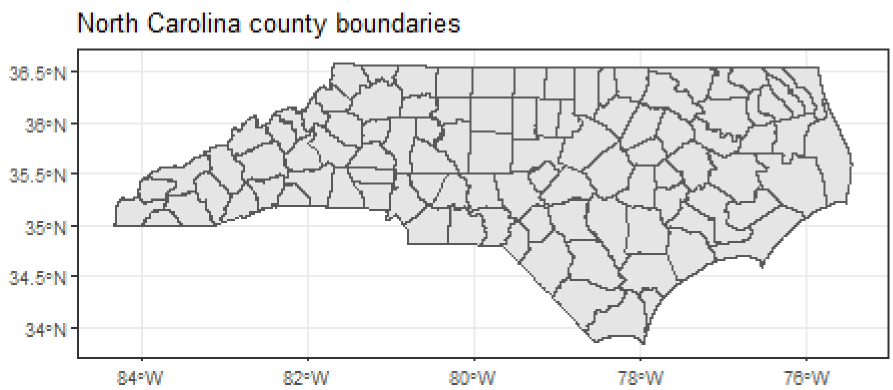
Spatial data are different

```r
library(sf)
nc <- st_read("https://opendata.arcgis.com/datasets/9728285994804c
                quiet = T)
nc[,1:3]
```

```
## Simple feature collection with 100 features and 3 fields
## geometry type:  POLYGON
## dimension:      XY
## bbox:           xmin: -84.32183 ymin: 33.8416 xmax: -75.45966 ymax: 36.
## geographic CRS: WGS 84
## First 10 features:
##    OBJECTID      County FIPS                       geometry
## 1         1      Camden   29 POLYGON ((-75.90629 36.0858...
## 2         2       Gates   73 POLYGON ((-76.69658 36.2961...
## 3         3     Iredell   97 POLYGON ((-80.94812 35.4911...
## 4         4      Wilkes  193 POLYGON ((-81.30257 36.0049...
## 5         5       Union  179 POLYGON ((-80.55036 35.2084...
## 6         6    Cabarrus   25 POLYGON ((-80.55036 35.2084...
## 7         7        Wake  183 POLYGON ((-78.90607 35.8681...
## 8         8    Franklin   69 POLYGON ((-78.25598 35.8181...
## 9         9      Pender  141 POLYGON ((-78.01193 34.7319...
## 10       10 New Hanover  129 POLYGON ((-77.71049 34.2979...
```

Basic plots

```
ggplot(nc) +
  geom_sf() +
  labs(title = "North Carolina county boundaries") +
  theme_bw()
```



Does anyone notice anything "weird" about this map?

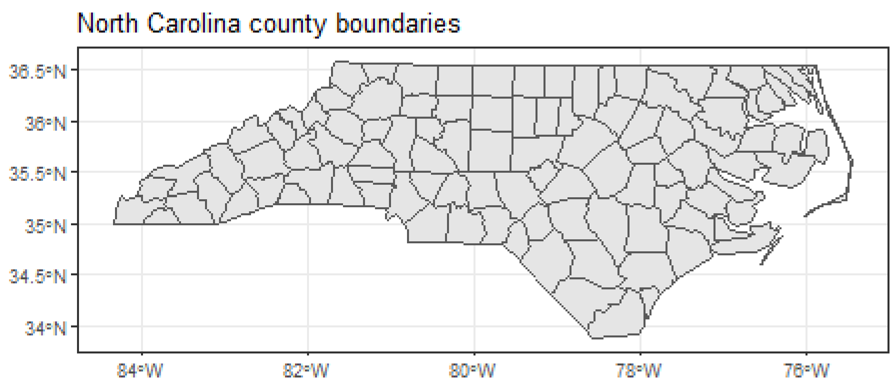
Basic plots

```
nc <- st_read("data/nc.shp", quiet = TRUE)
nc
```

```
## Simple feature collection with 100 features and 1 field
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax: 36
## geographic CRS: NAD27
## First 10 features:
##           name                       geometry
## 1         ASHE MULTIPOLYGON (((-81.47276 3...
## 2    ALLEGHANY MULTIPOLYGON (((-81.23989 3...
## 3        SURRY MULTIPOLYGON (((-80.45634 3...
## 4    CURRITUCK MULTIPOLYGON (((-76.00897 3...
## 5  NORTHAMPTON MULTIPOLYGON (((-77.21767 3...
## 6     HERTFORD MULTIPOLYGON (((-76.74506 3...
## 7       CAMDEN MULTIPOLYGON (((-76.00897 3...
## 8        GATES MULTIPOLYGON (((-76.56251 3...
## 9       WARREN MULTIPOLYGON (((-78.30876 3...
## 10      STOKES MULTIPOLYGON (((-80.02567 3...
```

Basic plots

```
ggplot(nc) +
  geom_sf() +
  labs(title = "North Carolina county boundaries") +
  theme_bw()
```



What's different about this map and shapefile?

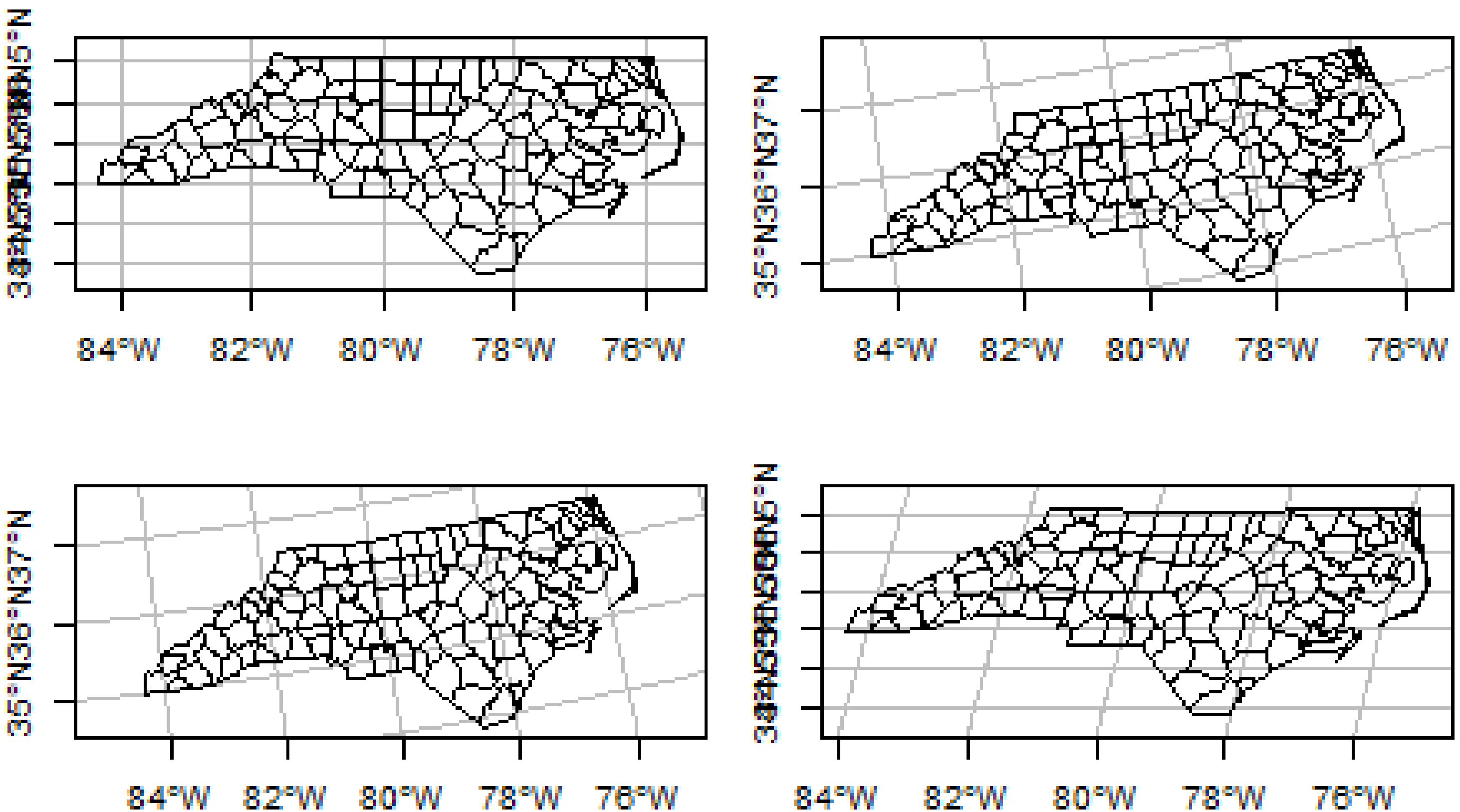
Basic plots

The geographic CRS and geometry were different (as was the file format).

Coordinate Reference System (CRS): defines how specific places are mapped onto a sptial map. **WGS 84** (used in the first file) is the latest revision of the World Geodetic System; **NAD27** (used in the second file) is the 1927 North American Datum.

Geometry: defines how the spatial object is "mapped"; the first file used a polygon geometry whereas the second file used a *multipolygon*.

Spatial data plotting needs care



Choropleth maps

When working with **areal** data, a **choropleth map** is a great visualization that colors regions by the value of some *numeric* value (as opposed to a simple coloring by category).

Let's create a choropleth map for the number of vaccines given out in NC using data from the NCDHHS (current as of March 15, 2021).

```
doses <- read_csv("data/nc_doses_031521.csv")
head(doses)
```

```
## # A tibble: 6 x 4
##   county    partial fully    pop
##   <chr>       <dbl> <dbl>  <dbl>
## 1 ALAMANCE    31945 19347 168761
## 2 ALEXANDER    6951  3976  38364
## 3 ALLEGHANY    2741  1927  11494
## 4 ANSON        4270  2677  23944
## 5 ASHE         5766  3247  27797
## 6 AVERY        4066  2667  18128
```

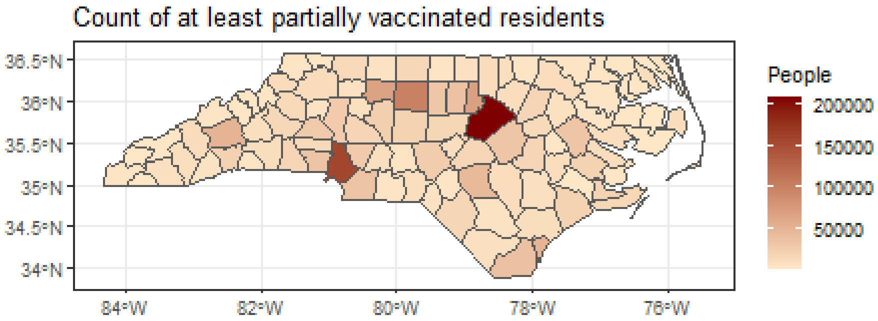
Choropleth maps

```r
doses <- rename(doses, name = county)
nc <- merge(nc, doses, by = "name")
nc
```

```
## Simple feature collection with 100 features and 4 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax: 36
## geographic CRS: NAD27
## First 10 features:
##         name partial fully    pop                       geometry
## 1   ALAMANCE   31945 19347 168761 MULTIPOLYGON (((-79.24619 3...
## 2  ALEXANDER    6951  3976  38364 MULTIPOLYGON (((-81.10889 3...
## 3  ALLEGHANY    2741  1927  11494 MULTIPOLYGON (((-81.23989 3...
## 4      ANSON    4270  2677  23944 MULTIPOLYGON (((-79.91995 3...
## 5       ASHE    5766  3247  27797 MULTIPOLYGON (((-81.47276 3...
## 6      AVERY    4066  2667  18128 MULTIPOLYGON (((-81.94135 3...
## 7   BEAUFORT   11493  7743  47436 MULTIPOLYGON (((-77.10377 3...
## 8     BERTIE    4701  2567  19630 MULTIPOLYGON (((-76.78307 3...
## 9     BLADEN    6653  4102  34475 MULTIPOLYGON (((-78.2615 34...
## 10 BRUNSWICK   38111 24790 143169 MULTIPOLYGON (((-78.65572 3...
```

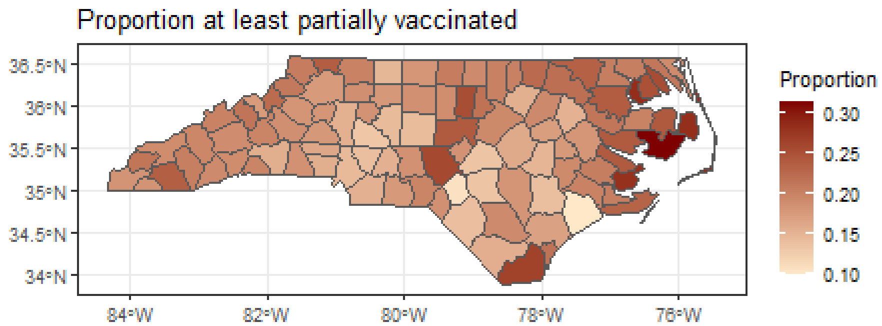
Choropleth maps

```
ggplot(nc) +
  geom_sf(aes(fill = partial)) +
  scale_fill_gradient(low = "#fee8c8", high = "#7f0000") +
  labs(title = "Count of at least partially vaccinated residents",
       fill = "People") +
  theme_bw()
```



Choropleth maps

```
ggplot(nc) +
  geom_sf(aes(fill = partial/pop)) +
  scale_fill_gradient(low = "#fee8c8", high = "#7f0000") +
  labs(title = "Proportion at least partially vaccinated",
       fill = "Proportion") +
  theme_bw()
```



Neighbors

Counties that are "close together" spatially might be similar.

One definition of "close" is to consider its **neighbors**. A neighbor is an observation that is **contiguous** to another (i.e., that shares a **border** or **point** in common). We can think of the **order** of contiguity as well: neighbors are first-order contiguous; neighbors-of-neighbors are second-order contiguous. Exact adjacency order depends on our definition (e.g., rook vs. queen on a chess board).

We might also consider observations to be "close" if they are within a certain distance of another observation. Distance-based adjacency measures are more commonly used with point data, while neighbor-based adjacency is more common with areal data.

Spatial weight matrix

A **spatial weight matrix** is a square matrix that identifies whether observations are neighbors (or more generally, the adjacency metric between all pairwise observations).

In general, it's pretty computationally-intensive to construct such a matrix, although built-in functions from R packages are pretty fast nowadays (and in this case we only have 100 counties).

```r
library(spdep)
sp_wts <- poly2nb(nc, row.names=nc$name, queen = T)
sp_wts
```

```
## Neighbour list object:
## Number of regions: 100
## Number of nonzero links: 490
## Percentage nonzero weights: 4.9
## Average number of links: 4.9
```

Spatial weight matrix

```
summary(sp_wts)
```

```
## Neighbour list object:
## Number of regions: 100
## Number of nonzero links: 490
## Percentage nonzero weights: 4.9
## Average number of links: 4.9
## Link number distribution:
##
##  2  3  4  5  6  7  8  9
##  8 15 17 23 19 14  2  2
## 8 least connected regions:
## 21 22 27 28 65 69 75 89 with 2 links
## 2 most connected regions:
## 49 63 with 9 links
```

Spatial weight matrix

```r
nc %>%
  slice(c(49, 63))
```

```
## Simple feature collection with 2 features and 4 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -81.10889 ymin: 35.03736 xmax: -79.09589 ymax: 36
## geographic CRS: NAD27
##      name partial fully    pop                       geometry
## 1 IREDELL   27949 18331 181071 MULTIPOLYGON (((-80.72652 3...
## 2   MOORE   26319 15706 101219 MULTIPOLYGON (((-79.60747 3...
```

Spatial weight matrix

```
sp_mat <- nb2mat(sp_wts, style='B') # Binary 1/0
sp_mat[1:10,1:10]
```

```
##    [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## 1     0    0    0    0    0    0    0    0    0     0
## 2     0    0    0    0    0    0    0    0    0     0
## 3     0    0    0    0    1    0    0    0    0     0
## 4     0    0    0    0    0    0    0    0    0     0
## 5     0    0    1    0    0    0    0    0    0     0
## 6     0    0    0    0    0    0    0    0    0     0
## 7     0    0    0    0    0    0    0    0    0     0
## 8     0    0    0    0    0    0    0    0    0     0
## 9     0    0    0    0    0    0    0    0    0     0
## 10    0    0    0    0    0    0    0    0    0     0
```

Spatial weight matrix

```
nc %>%
  slice(which(sp_mat[which(nc$name == "DURHAM"),] == 1)) %>%
  select(name) %>%
  st_drop_geometry()
```

What does the above code do?

Spatial weight matrix

```
nc %>%
  slice(which(sp_mat[which(nc$name == "DURHAM"),] > 0)) %>%
  select(name) %>%
  st_drop_geometry()
```

Spatial weight matrix

```
cbind(nc, neighbors = rowSums(sp_mat))
```

```
## Simple feature collection with 100 features and 5 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax: 36
## geographic CRS: NAD27
## First 10 features:
##         name partial fully    pop neighbors                       geome
## 1   ALAMANCE   31945 19347 168761         6 MULTIPOLYGON (((-79.24619 3
## 2  ALEXANDER    6951  3976  38364         4 MULTIPOLYGON (((-81.10889 3
## 3  ALLEGHANY    2741  1927  11494         3 MULTIPOLYGON (((-81.23989 3
## 4      ANSON    4270  2677  23944         4 MULTIPOLYGON (((-79.91995 3
## 5       ASHE    5766  3247  27797         3 MULTIPOLYGON (((-81.47276 3
## 6      AVERY    4066  2667  18128         5 MULTIPOLYGON (((-81.94135 3
## 7   BEAUFORT   11493  7743  47436         6 MULTIPOLYGON (((-77.10377 3
## 8     BERTIE    4701  2567  19630         5 MULTIPOLYGON (((-76.78307 3
## 9     BLADEN    6653  4102  34475         5 MULTIPOLYGON (((-78.2615 34
## 10 BRUNSWICK   38111 24790 143169         3 MULTIPOLYGON (((-78.65572 3
```

Spatial weight matrix

```
sp_mat_std <- nb2mat(sp_wts, style='W') # Row-standardized
sp_mat_std[1:10,1:10]
```

```
##    [,1] [,2]      [,3] [,4]      [,5] [,6] [,7] [,8] [,9] [,10]
## 1     0    0 0.0000000    0 0.0000000    0    0    0    0     0
## 2     0    0 0.0000000    0 0.0000000    0    0    0    0     0
## 3     0    0 0.0000000    0 0.3333333    0    0    0    0     0
## 4     0    0 0.0000000    0 0.0000000    0    0    0    0     0
## 5     0    0 0.3333333    0 0.0000000    0    0    0    0     0
## 6     0    0 0.0000000    0 0.0000000    0    0    0    0     0
## 7     0    0 0.0000000    0 0.0000000    0    0    0    0     0
## 8     0    0 0.0000000    0 0.0000000    0    0    0    0     0
## 9     0    0 0.0000000    0 0.0000000    0    0    0    0     0
## 10    0    0 0.0000000    0 0.0000000    0    0    0    0     0
```

Moran's I

```
sp_mat_list <-  nb2listw(sp_wts, style='B')
sp_mat_list
```

```
## Characteristics of weights list object:
## Neighbour list object:
## Number of regions: 100
## Number of nonzero links: 490
## Percentage nonzero weights: 4.9
## Average number of links: 4.9
##
## Weights style: B
## Weights constants summary:
##     n    nn  S0  S1    S2
## B 100 10000 490 980 10696
```

Moran's I

$$I = \frac{n}{\sum_i \sum_j w_{ij}} \frac{\sum_i \sum_j w_{ij}(y_i - \bar{y})(y_j - \bar{y})}{\sum_i (y_i - \bar{y})^2}$$

- n is the number of spatial observations
- w_{ij} is the spatial weight between spatial observations i and j

I thus depends on how we constructed our spatial weight matrix. We've shown a binary weight matrix and its row-standardized version, but we can also include information regarding how much border they share, or distance between centroids, etc.

Moran's I

```
moran(nc$partial/nc$pop, sp_mat_list, nrow(nc), sum(sp_mat))
```

```
## $I
## [1] 0.3201586
##
## $K
## [1] 3.586839
```

Positive I suggests spatial clustering - that higher values are "close" to other higher values, and lower values are "close" to other lower values. Negative I suggests spatial dispersion - that higher values are "close" to lower values, and vice-versa.

Moran's I

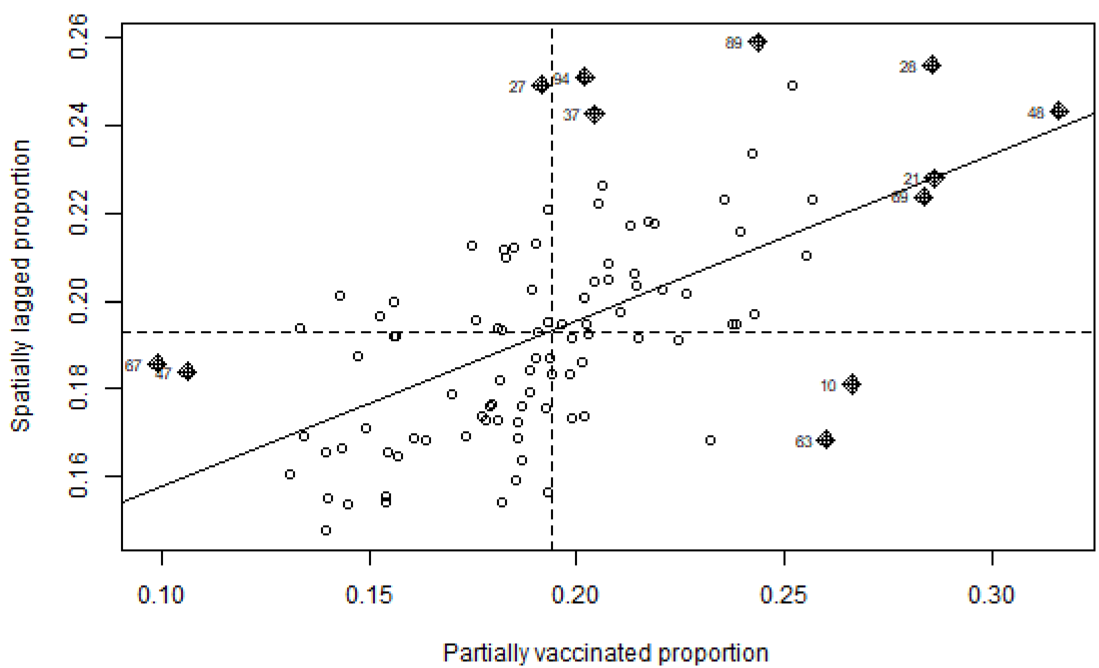
```
set.seed(123)
moran.mc(nc$partial/nc$pop, sp_mat_list, nsim = 999)
```

```
##
##      Monte-Carlo simulation of Moran I
##
## data:  nc$partial/nc$pop
## weights: sp_mat_list
## number of simulations + 1: 1000
##
## statistic = 0.32016, observed rank = 1000, p-value = 0.001
## alternative hypothesis: greater
```

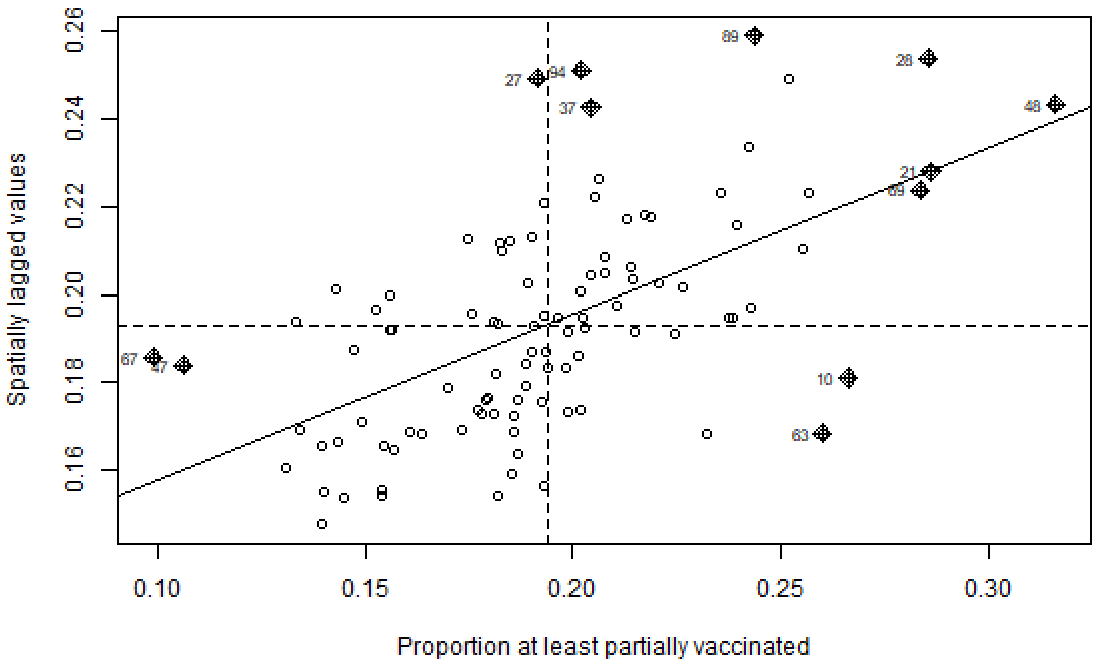
What might we conclude regarding proportion of residents receiving at least one dose of the vaccine?

Moran's I

```
sp_mat_list_std <-  nb2listw(sp_wts, style='W')
moran.plot(nc$partial/nc$pop, sp_mat_list_std,
           xlab = "Partially vaccinated proportion",
           ylab = "Spatially lagged proportion")
```



Moran's I



What counties have a relatively high proportion of at least partially vaccinated people, but are surrounded by less-vaccinated counties?

Moran's I

```
nc %>%
  slice(c(63)) %>%
  mutate(prop = partial/pop) %>%
  select(name, prop) %>%
  st_drop_geometry
```

```
##    name      prop
## 1 MOORE 0.2600204
```

Moran's I

```
nc %>%
  slice(which(sp_mat[63,] > 0)) %>%
  mutate(prop = partial/pop) %>%
  select(name, prop) %>%
  st_drop_geometry()
```

```
##         name      prop
## 1    CHATHAM 0.2428902
## 2 CUMBERLAND 0.1343745
## 3    HARNETT 0.1330185
## 4       HOKE 0.1060106
## 5        LEE 0.1851131
## 6 MONTGOMERY 0.1857236
## 7   RANDOLPH 0.1426872
## 8   RICHMOND 0.1985017
## 9   SCOTLAND 0.1867923
```

How might we further account for population size when determining proportion of (at least partially) vaccinated neighbors?