

# Spatial data & visualization

Prof. Maria Tackett





STA 199

**Click for PDF of slides**



# Introduction

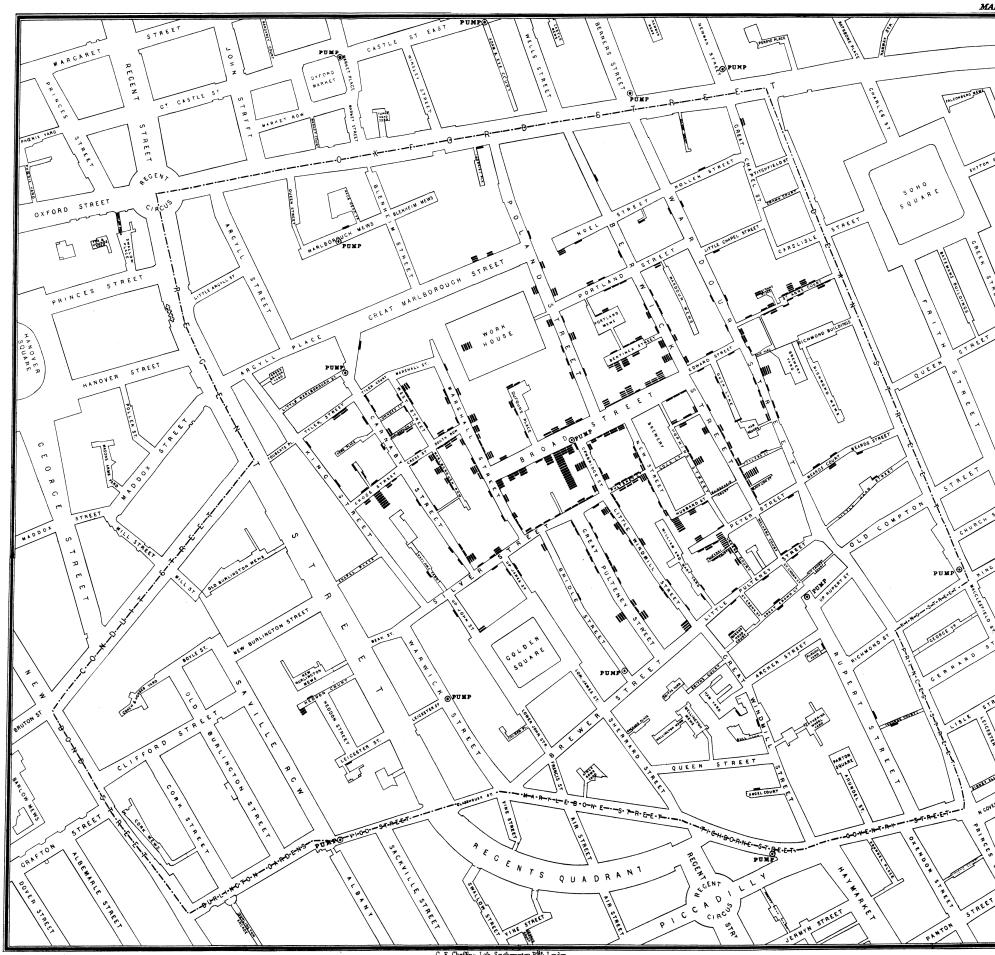


# Spatial data is important

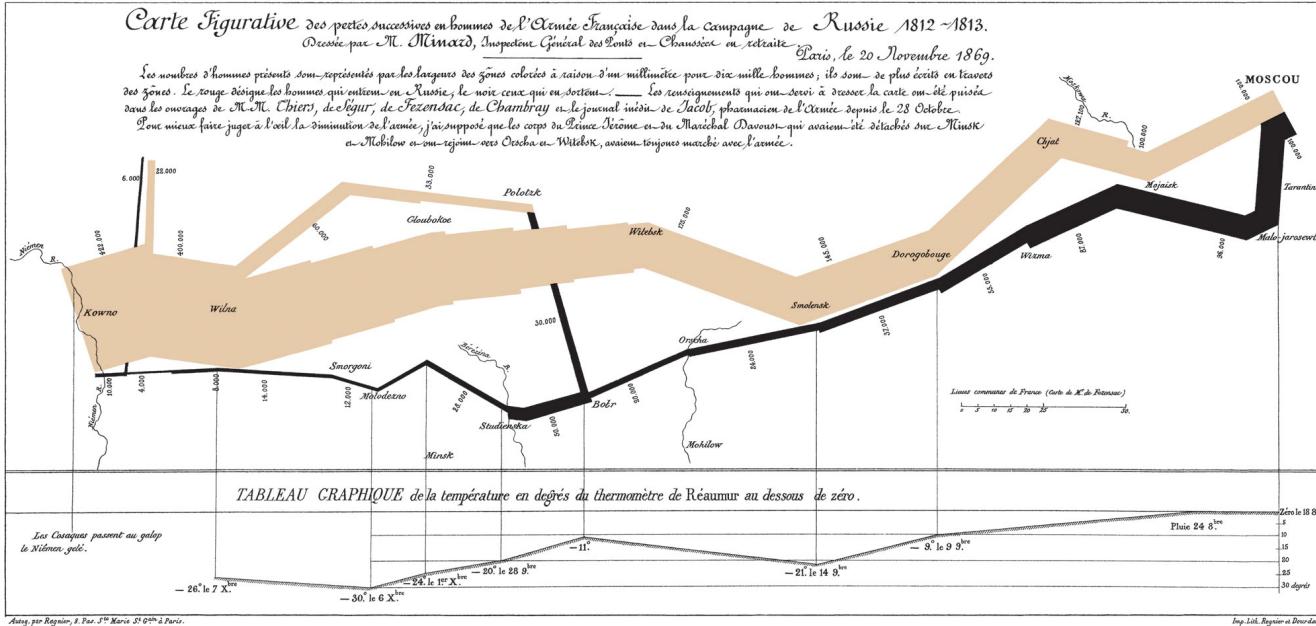
- exploratory data analysis
- detecting spatial patterns and trends
- understanding spatial data relationships
- analysis of spatial data should reflect spatial structure



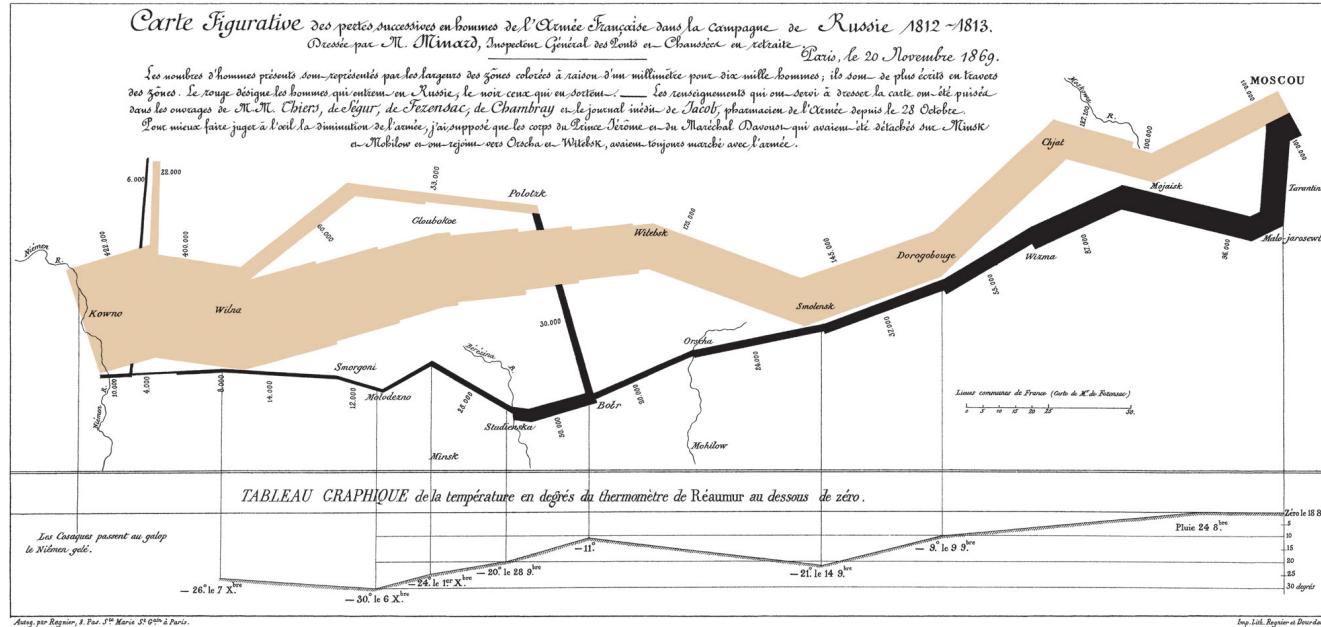
# 1854 London Cholera Outbreak



# Napoleon's 1812 Russia Campaign



# Napoleon's 1812 Russia Campaign



Many others!

- Migrations
- World Population Density
- Global Power

# Spatial data is different

Our typical tidy data frame:

```
## # A tibble: 336,776 x 19
##   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>     <int>          <int>     <dbl>     <int>          <int>
## 1 2013     1     1      517            515        2       830          819
## 2 2013     1     1      533            529        4       850          830
## 3 2013     1     1      542            540        2       923          850
## 4 2013     1     1      544            545       -1      1004         1022
## 5 2013     1     1      554            600       -6      812          837
## 6 2013     1     1      554            558       -4      740          728
## 7 2013     1     1      555            600       -5      913          854
## 8 2013     1     1      557            600       -3      709          723
## 9 2013     1     1      557            600       -3      838          846
## 10 2013    1     1      558            600       -2      753          745
## # ... with 336,766 more rows, and 11 more variables: arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
```



# Spatial data is different

Our (new) simple feature object:

```
## Simple feature collection with 100 features and 3 fields
## geometry type: MULTIPOLYGON
## dimension: XY
## bbox: xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax: 36.58965
## geographic CRS: NAD27
## First 10 features:
##           name regstrd voted               geometry
## 1      ASHE    19414   8428 MULTIPOLYGON ((((-81.47276 3...
## 2 ALLEGHANY     7556   4101 MULTIPOLYGON ((((-81.23989 3...
## 3      SURRY    46666  23660 MULTIPOLYGON ((((-80.45634 3...
## 4 CURREITUCK    21803   7536 MULTIPOLYGON ((((-76.00897 3...
## 5 NORTHAMPTON   13891   6196 MULTIPOLYGON ((((-77.21767 3...
## 6 HERTFORD     14945   6955 MULTIPOLYGON ((((-76.74506 3...
## 7      CAMDEN     8128   3472 MULTIPOLYGON ((((-76.00897 3...
## 8       GATES     8294   3105 MULTIPOLYGON ((((-76.56251 3...
## 9      WARREN    13441   6878 MULTIPOLYGON ((((-78.30876 3...
## 10     STOKES    31649  14444 MULTIPOLYGON ((((-80.02567 3...
```

# Spatial data is different

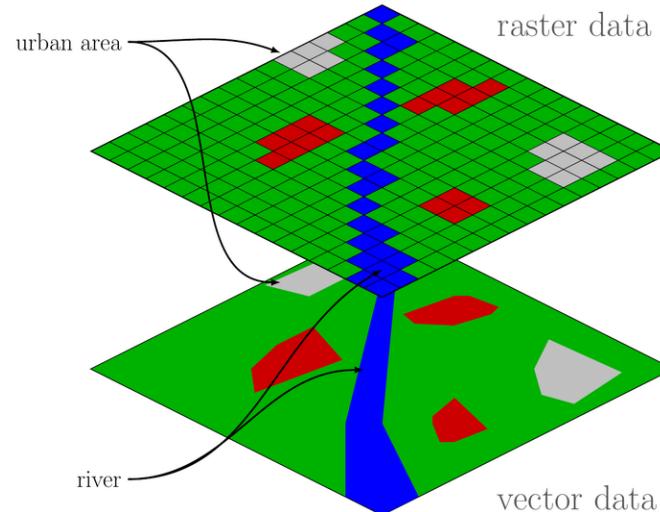
Our (new) simple feature object:

```
## Simple feature collection with 100 features and 3 fields
## geometry type: MULTIPOLYGON
## dimension: XY
## bbox: xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax: 36.58965
## geographic CRS: NAD27
## First 10 features:
##           name regstrd voted               geometry
## 1      ASHE    19414   8428 MULTIPOLYGON ((((-81.47276 3...
## 2 ALLEGHANY     7556   4101 MULTIPOLYGON ((((-81.23989 3...
## 3      SURRY    46666  23660 MULTIPOLYGON ((((-80.45634 3...
## 4 CURREITUCK    21803   7536 MULTIPOLYGON ((((-76.00897 3...
## 5 NORTHAMPTON   13891   6196 MULTIPOLYGON ((((-77.21767 3...
## 6 HERTFORD     14945   6955 MULTIPOLYGON ((((-76.74506 3...
## 7      CAMDEN     8128   3472 MULTIPOLYGON ((((-76.00897 3...
## 8       GATES     8294   3105 MULTIPOLYGON ((((-76.56251 3...
## 9      WARREN    13441   6878 MULTIPOLYGON ((((-78.30876 3...
## 10     STOKES    31649  14444 MULTIPOLYGON ((((-80.02567 3...
```

# Raster versus vector spatial data

**Vector** spatial data describes the world using shapes (points, lines, polygons, etc).

**Raster** spatial data describes the world using cells of constant size.



The choice to use vector or raster data depends on the problem context. We will focus on **vector** data.

# Simple features

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.



# Simple features

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.

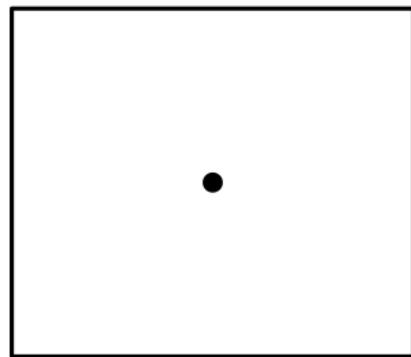
The package **sf** implements simple features and other spatial functionality using **tidy** principles.



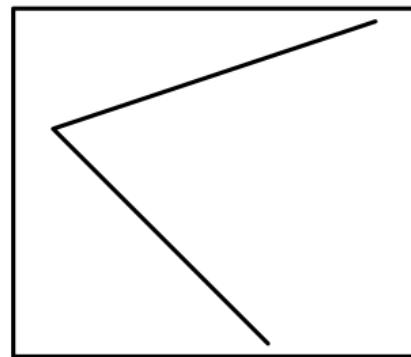
# Simple features

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

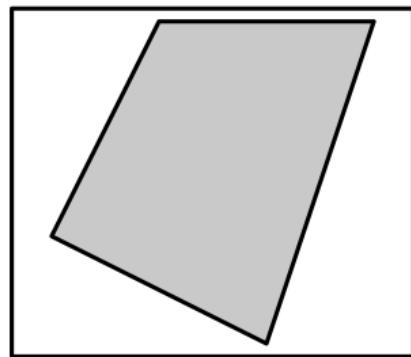
**Point**



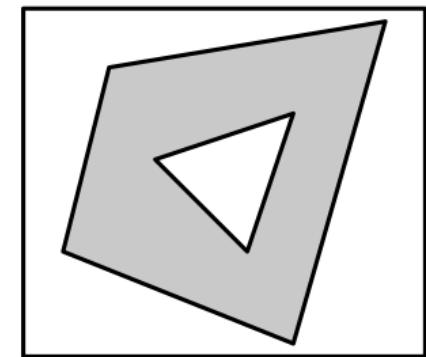
**Linestring**



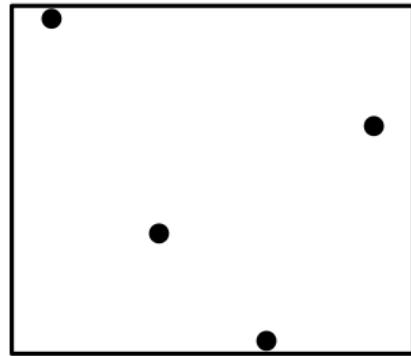
**Polygon**



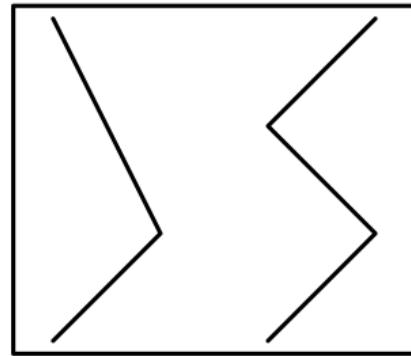
**Polygon w/ Hole(s)**



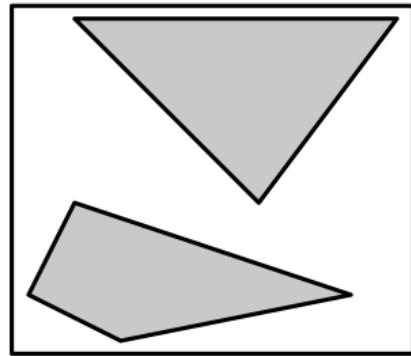
**Multipoint**



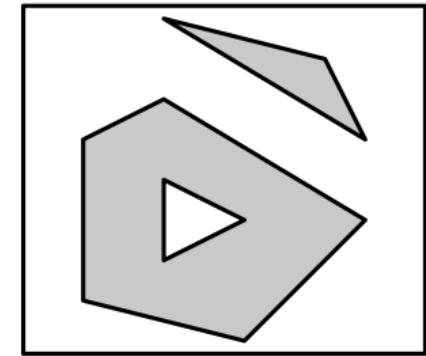
**Multilinestring**



**Multipolygon**



**Multipolygon w/ Hole(**



# A simple feature object

- Simple features are stored in a data frame, with the geographic information in a column called **geometry**.
- Simple features can contain both spatial and non-spatial data.
- Functions for spatial data in **sf** begin **st\_**.



# A simple feature object

```
## Simple feature collection with 100 features and 6 fields
## geometry type: MULTIPOLYGON
## dimension: XY
## bbox: xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax: 36.58965
## geographic CRS: NAD27
## First 10 features:
##           name regstrd voted mailed rejectd ml_rqst
## 1      ASHE    19414   8428     NA     NA    2666
## 2 ALLEGHANY     7556   4101     NA     NA     971
## 3      SURRY    46666  23660   4366      7    7088
## 4 CURRITUCK    21803   7536     NA     NA    2472
## 5 NORTHAMPTON   13891   6196     828      2    1441
## 6 HERTFORD     14945   6955     NA     NA    1524
## 7      CAMDEN    8128   3472     416      1     739
## 8      GATES    8294   3105     NA     NA     847
## 9      WARREN   13441   6878     NA     NA    1913
## 10     STOKES   31649  14444   2162      2    3648
##                               geometry
## 1 MULTIPOLYGON (((-81.47276 3...
## 2 MULTIPOLYGON (((-81.23989 3...
## 3 MULTIPOLYGON (((-80.45634 3...
## 4 MULTIPOLYGON ((((-76.00897 3...
```

# Visualizing spatial data



# nc\_votes

This data was pulled from the [North Carolina Early Voting Statistics](#) website and is current as of 10-28-2020.

The dataset contains the following variables:

- **name**: county name
- **regstrd**: number of registered voters
- **voted**: number of individuals who have voted
- **mailed**: number of mail ballots returned
- **rejectd**: number of mail ballots rejected
- **ml\_rqst**: number of mail ballots requested

# Getting `sf` objects

To read simple features from a file or database use the function `st_read()`.

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

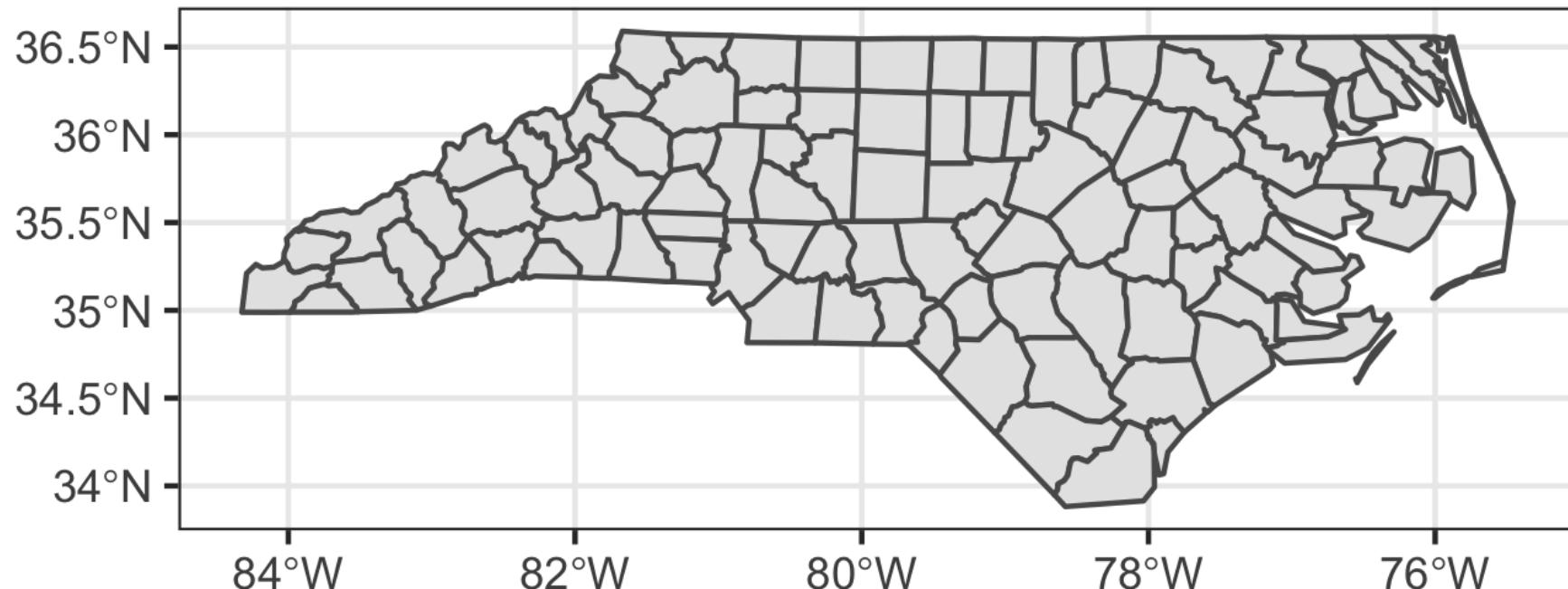
## Simple feature collection with 100 features and 6 fields
## geometry type: MULTIPOLYGON
## dimension: XY
## bbox:           xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax: 36.58965
## geographic CRS: NAD27
## First 10 features:
##          name regstrd voted mailed rejectd ml_rqst
## 1      ASHE   19414    8428     NA     NA    2666
## 2  ALLEGHANY    7556    4101     NA     NA     971
## 3      SURRY   46666   23660    4366      7    7088
## 4  CURRITUCK   21803    7536     NA     NA    2472
## 5 NORTHAMPTON   13891    6196    828      2    1441
## 6   HERTFORD   14945    6955     NA     NA    1524
## 7     CAMDEN    8128    3472    416      1     739
## 8      GATES   8294    3105     NA     NA     847
## 9     WARREN   13441    6878     NA     NA    1913
## 10    STOKES   31649   14444    2162      2    3648
##          geometry
```



# Plotting with ggplot()

```
ggplot(nc) +  
  geom_sf() +  
  labs(title = "North Carolina counties")
```

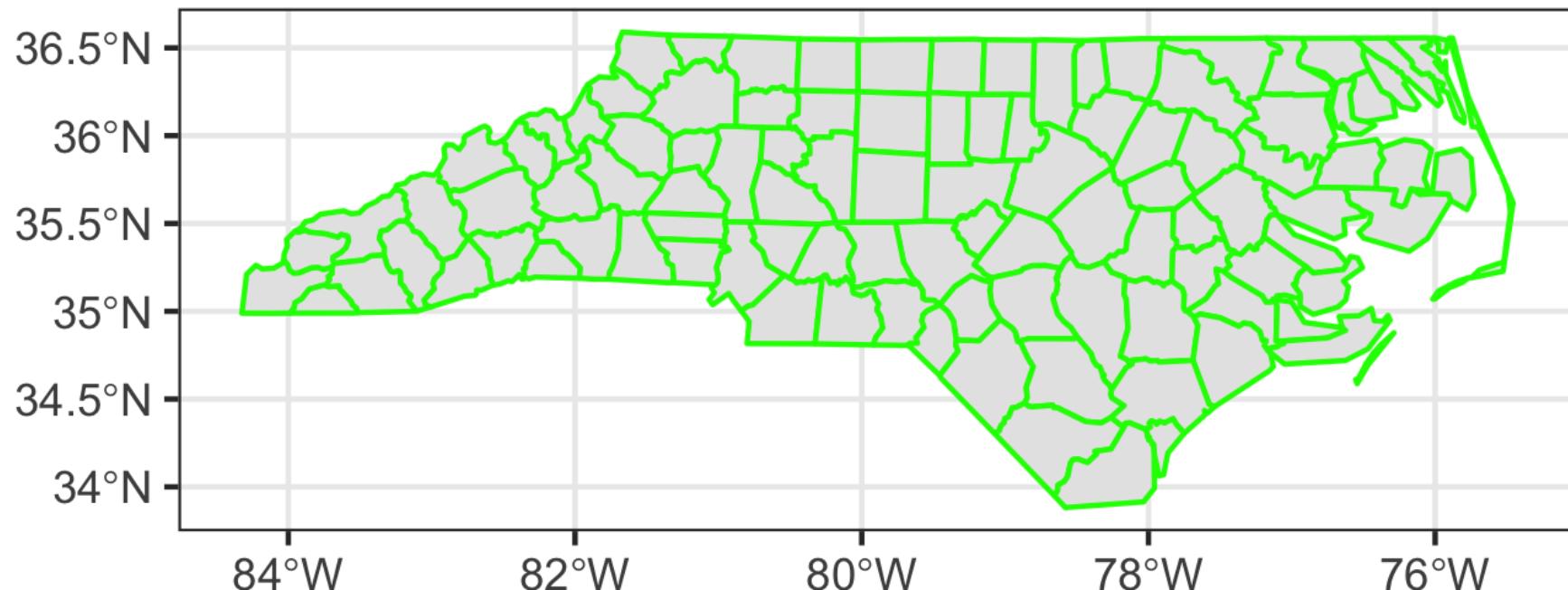
North Carolina counties



# A look at some aesthetics

```
ggplot(nc) +  
  geom_sf(color = "green") +  
  labs(title = "North Carolina counties with theme and aesthetics")
```

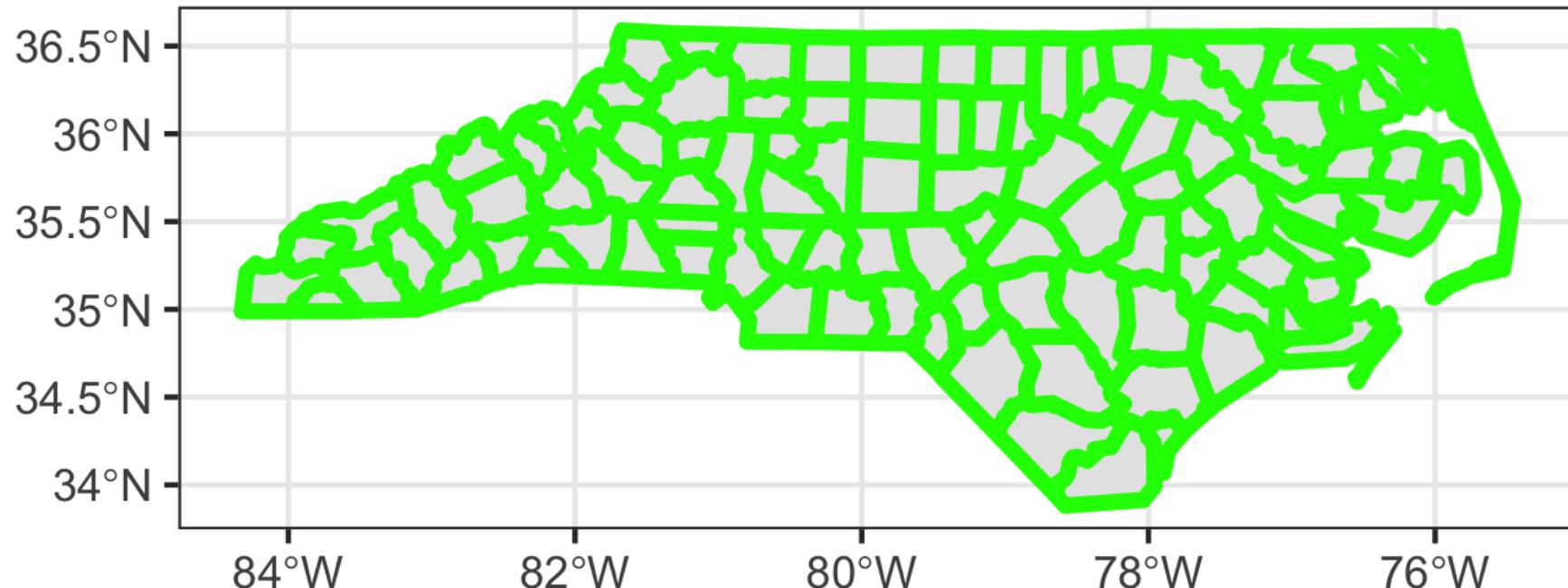
North Carolina counties with theme and aesthetics



# A look at some aesthetics

```
ggplot(nc) +  
  geom_sf(color = "green", size = 1.5) +  
  labs(title = "North Carolina counties with theme and aesthetics")
```

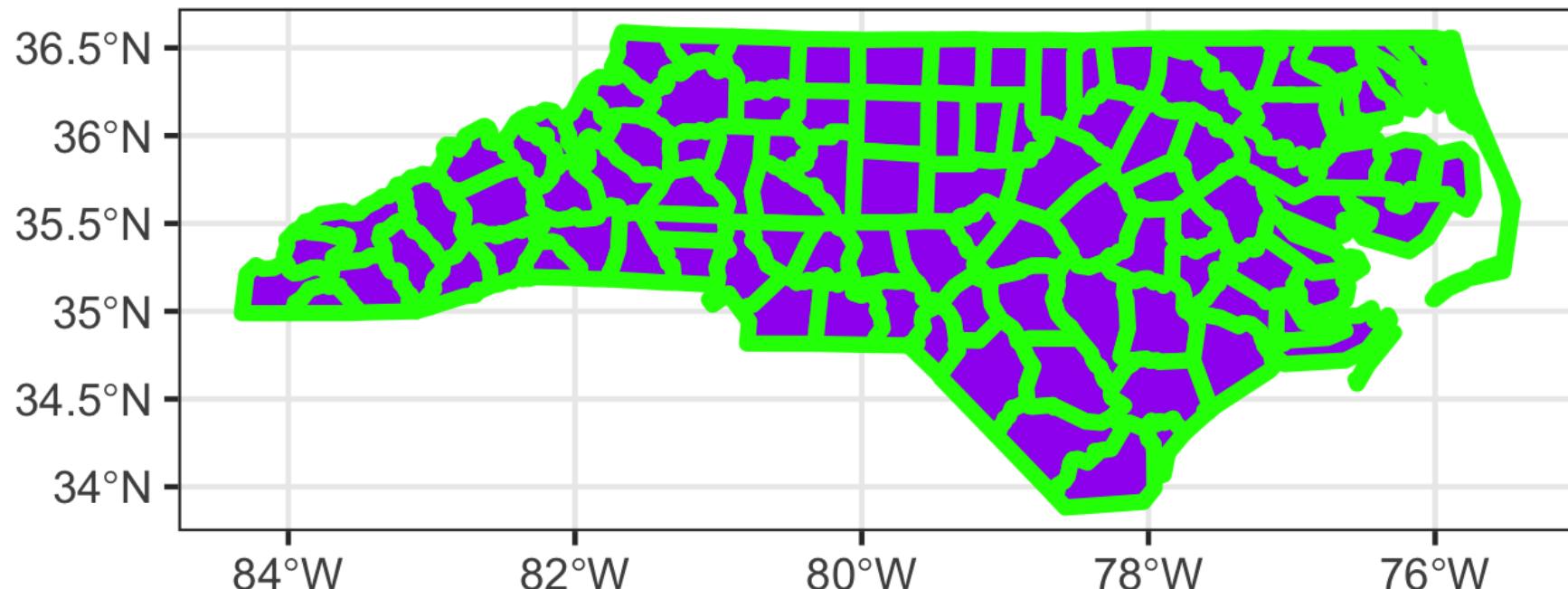
North Carolina counties with theme and aesthetics



# A look at some aesthetics

```
ggplot(nc) +  
  geom_sf(color = "green", size = 1.5, fill = "purple") +  
  labs(title = "North Carolina counties with theme and aesthetics")
```

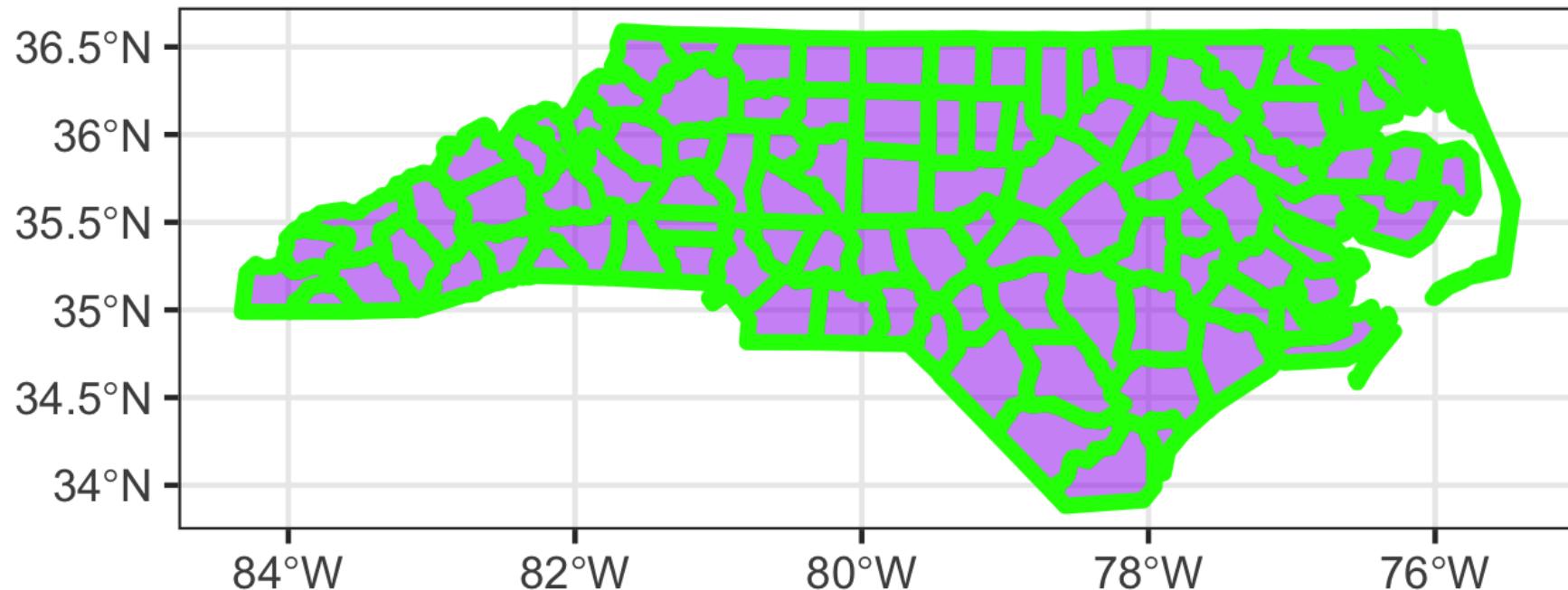
North Carolina counties with theme and aesthetics



# A look at some aesthetics

```
ggplot(nc) +  
  geom_sf(color = "green", size = 1.5, fill = "purple", alpha = 0.50)  
  labs(title = "North Carolina counties with theme and aesthetics")
```

North Carolina counties with theme and aesthetics



# A look back at some of our data

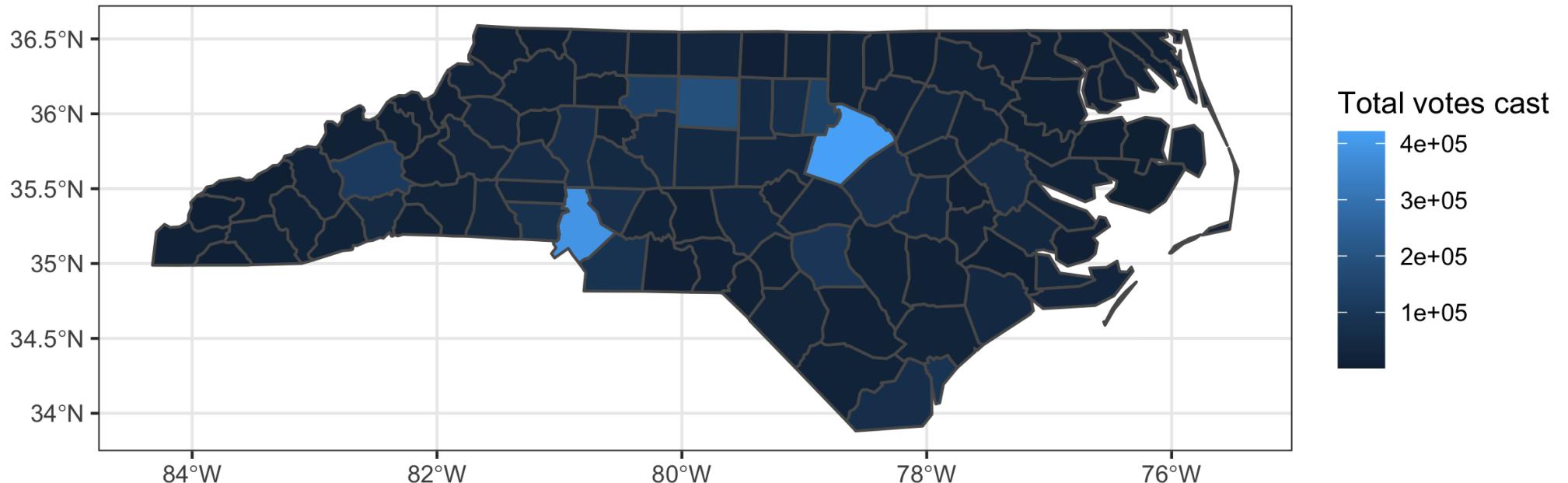
```
## Simple feature collection with 100 features and 6 fields
## geometry type: MULTIPOLYGON
## dimension: XY
## bbox: xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax: 36.58965
## geographic CRS: NAD27
## First 10 features:
##           name regstrd voted mailed rejectd ml_rqst
## 1        ASHE  19414   8428     NA     NA    2666
## 2 ALLEGHANY    7556   4101     NA     NA     971
## 3      SURRY  46666  23660   4366      7   7088
## 4 CURRITUCK  21803   7536     NA     NA   2472
## 5 NORTHAMPTON 13891   6196   828       2   1441
## 6 HERTFORD  14945   6955     NA     NA   1524
## 7      CAMDEN   8128   3472   416       1    739
## 8      GATES   8294   3105     NA     NA    847
## 9      WARREN  13441   6878     NA     NA   1913
## 10     STOKES  31649  14444  2162       2   3648
##               geometry
## 1 MULTIPOLYGON (((-81.47276 3...
## 2 MULTIPOLYGON (((-81.23989 3...
## 3 MULTIPOLYGON (((-80.45634 3...
## 4 MULTIPOLYGON (((-76.00897 3...
## 5 MULTIPOLYGON (((-77.21767 3...
## 6 MULTIPOLYGON (((-76.74506 3...
## 7 MULTIPOLYGON (((-76.00897 3...
## 8 MULTIPOLYGON (((-76.56251 3...
```



# Choropleth map

Plot    Code

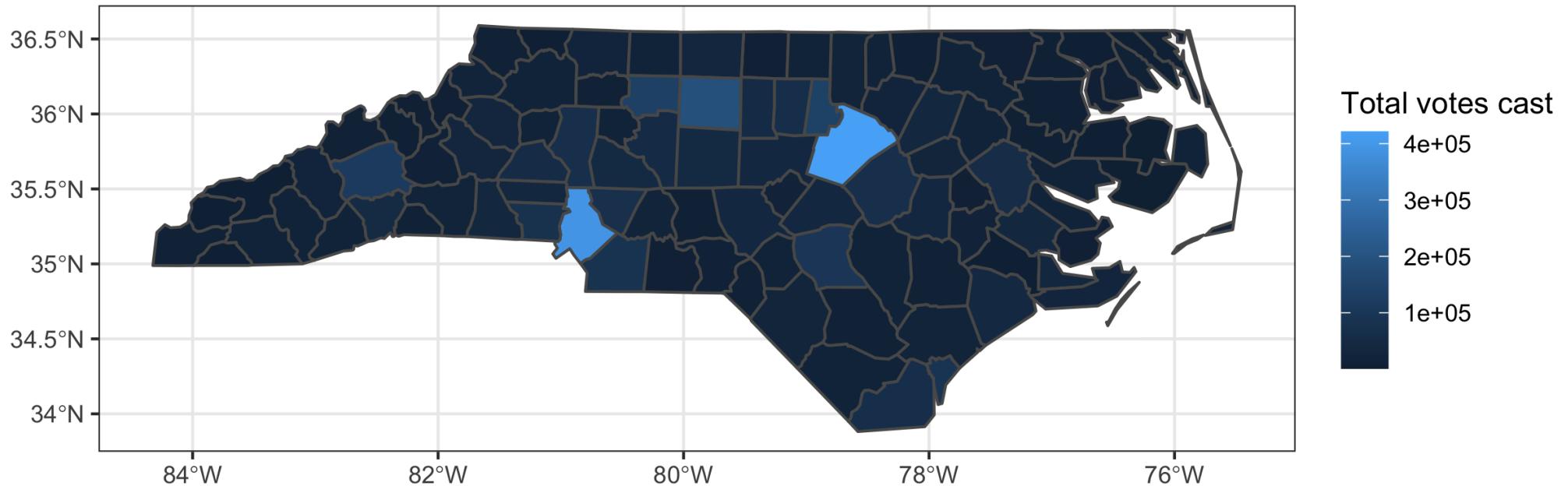
Higher population counties have more votes cast



# Choropleth map

Plot    Code

Higher population counties have more votes cast



It is sometimes helpful to pick diverging colors, [colorbrewer2](#) can help.

# Choropleth map

One way to set fill colors is with `scale_fill_gradient()`.

---

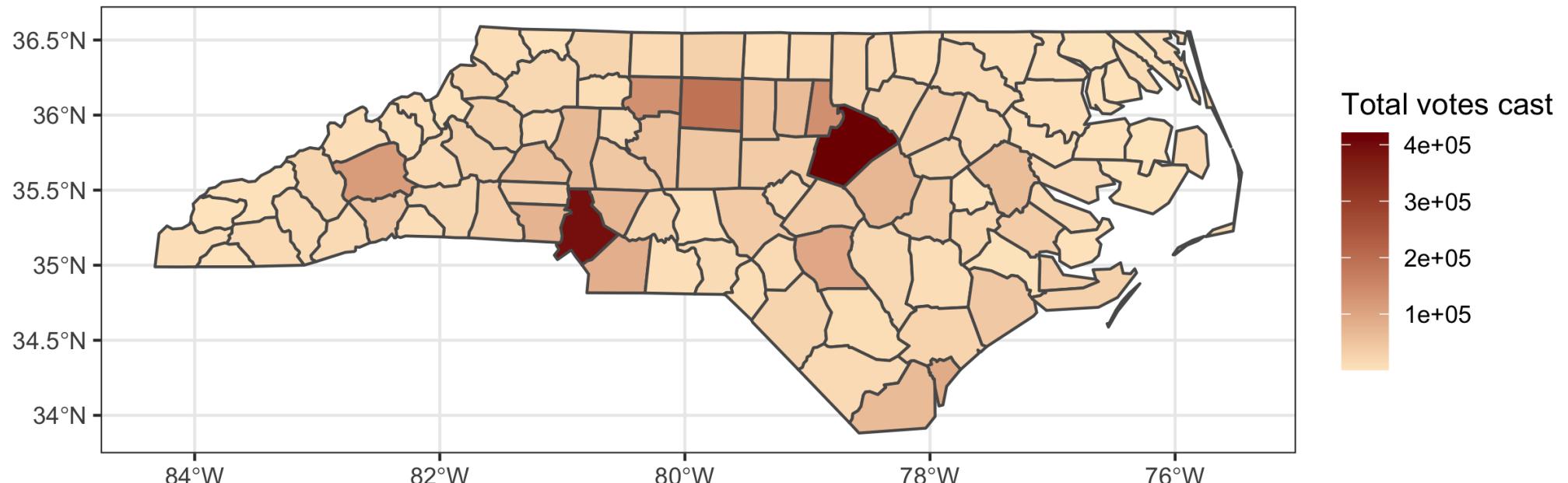
Plot

---

Code

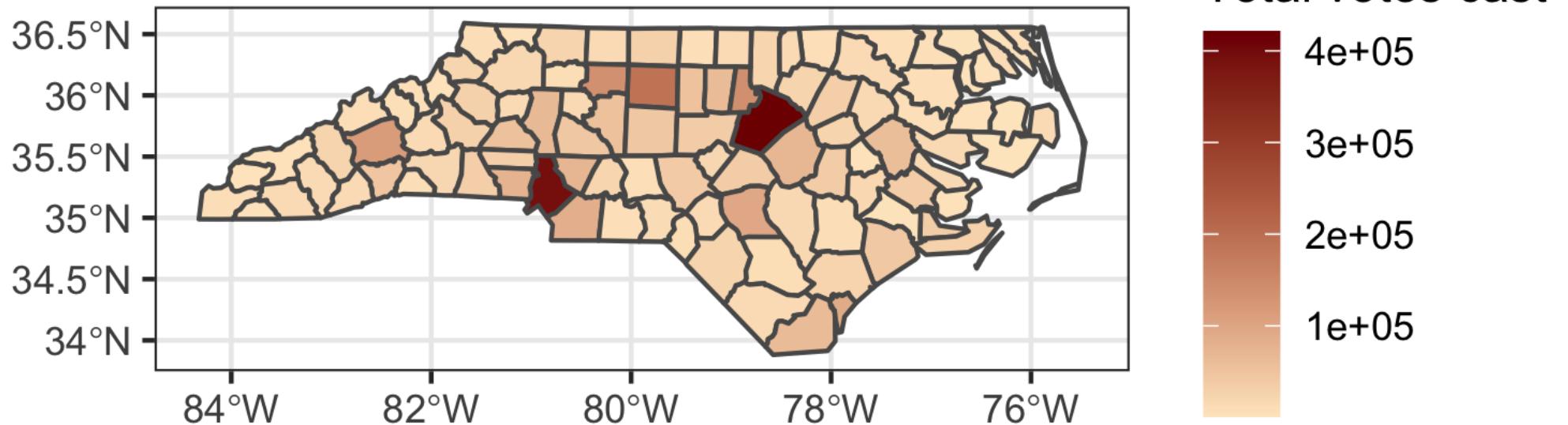
---

Higher population counties have more votes cast



# "...it's just a population map!"

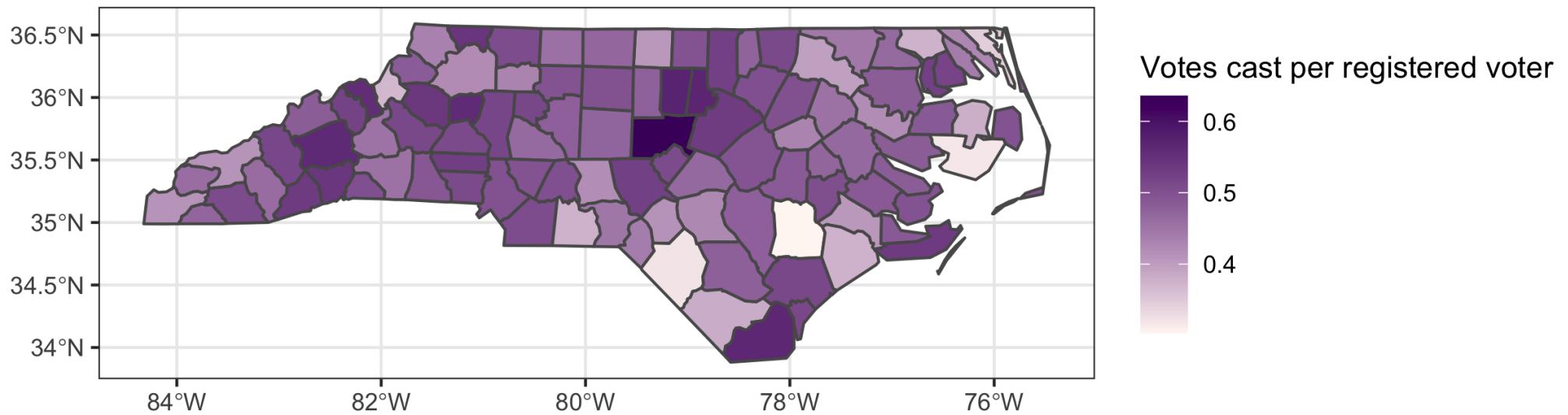
Higher population counties have more votes cast



# Let's make it more informative

Plot    Code

Early vote turnout varies by county



# Map layers

# Game Lands data

The North Carolina Department of Environment and Natural Resources, Wildlife Resources Commission and the NC Center for Geographic Information and Analysis has a **shapefile data set** available on all public Game Lands in NC.

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



# A closer look

```
nc_game
```

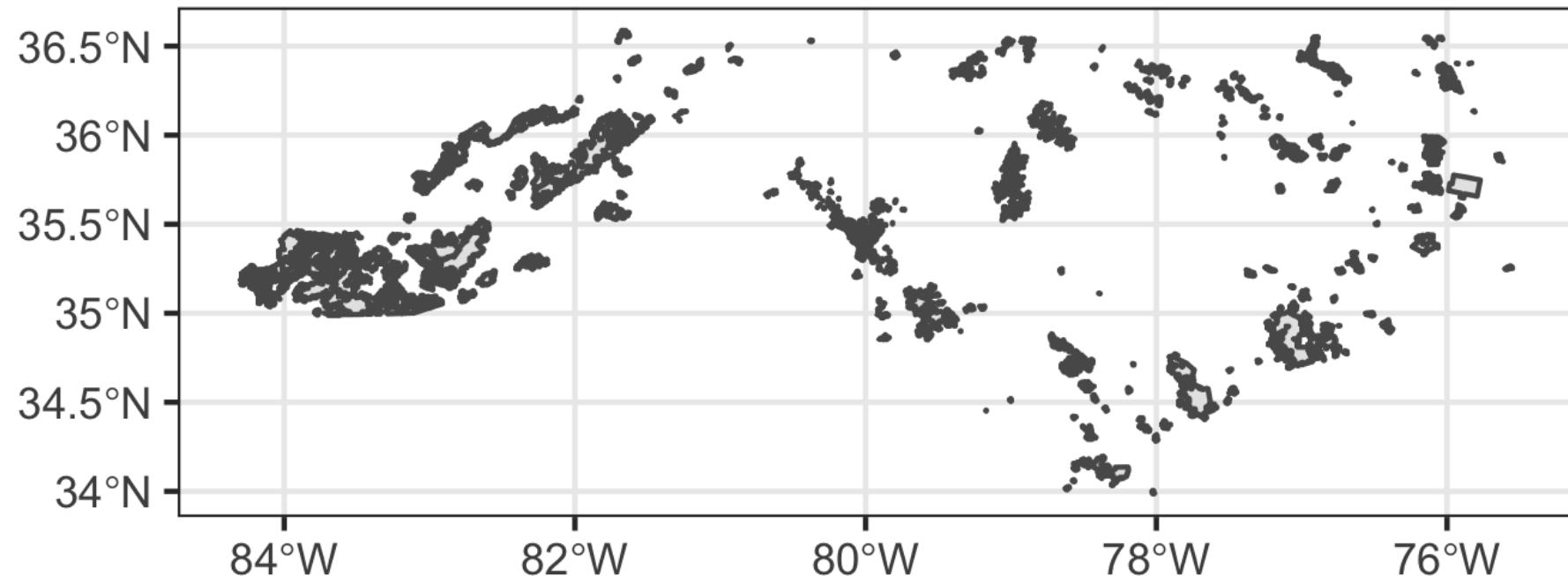
```
## Simple feature collection with 94 features and 6 fields
## geometry type: MULTIPOLYGON
## dimension: XY
## bbox: xmin: -84.29534 ymin: 33.98542 xmax: -75.54947 ymax: 36.58814
## geographic CRS: NAD27
## First 10 features:
##   OBJECTID          GML_HAB SUM_ACRES GameLandID Shape__Are
## 1      1             Alcoa  11395.9471        1    69931121
## 2      2       Alligator River  24439.0891        2   151120825
## 3      3           Angola Bay  34063.4468        3  204400526
## 4      4         Bachelor Bay  2786.2577        4   17219484
## 5      5        Bertie County  3883.7683        5  24044312
## 6      6 Bladen Lakes State Forest  33671.8426        6  202085696
## 7      7      Brinkleyville  1843.8439       92  11511489
## 8      8            Buckhorn  491.3477       81   3046371
## 9      9          Buckridge  17965.7187      10  110580903
## 10     10        Buffalo Cove  6630.9453      11  41161465
##   Shape__Len           geometry
## 1 549030.42 MULTIPOLYGON (((-80.07347 3...
```



# Visualize nc\_game

```
ggplot(nc_game) +  
  geom_sf() +  
  labs(title = "North Carolina gamelands")
```

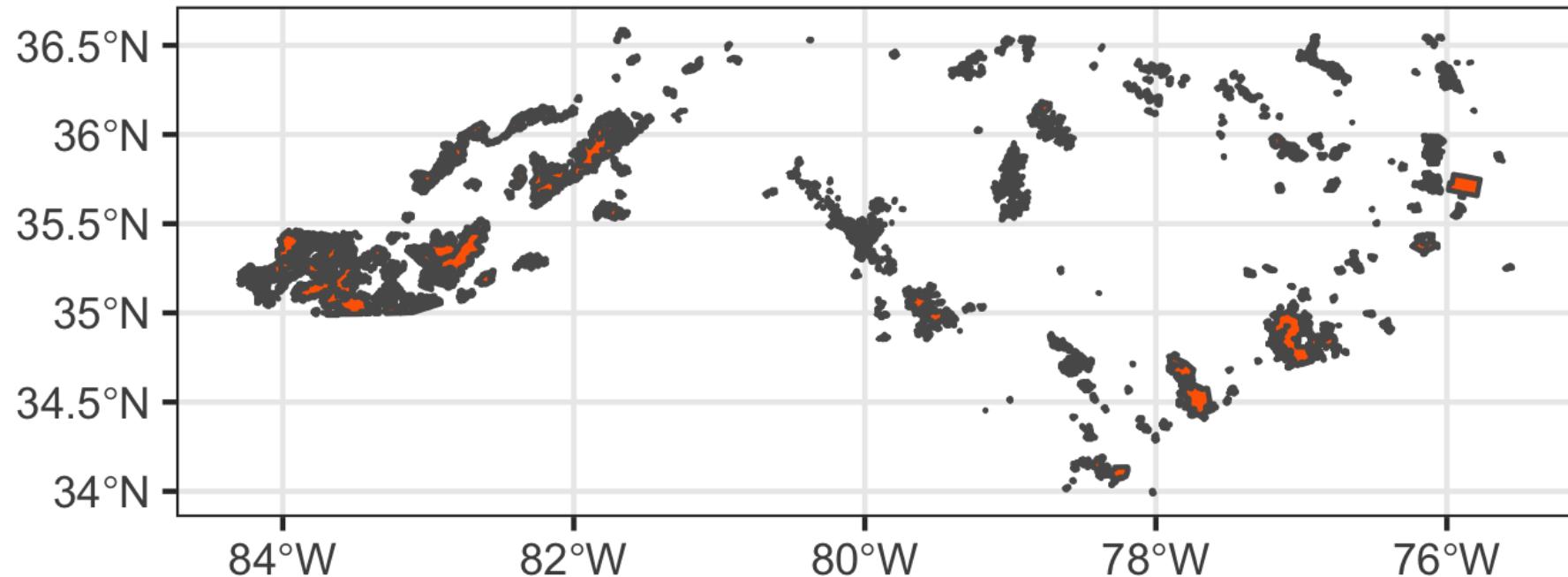
North Carolina gamelands



# Visualize nc\_game

```
ggplot(nc_game) +  
  geom_sf(fill = "#ff6700") +  
  labs(title = "North Carolina gamelands")
```

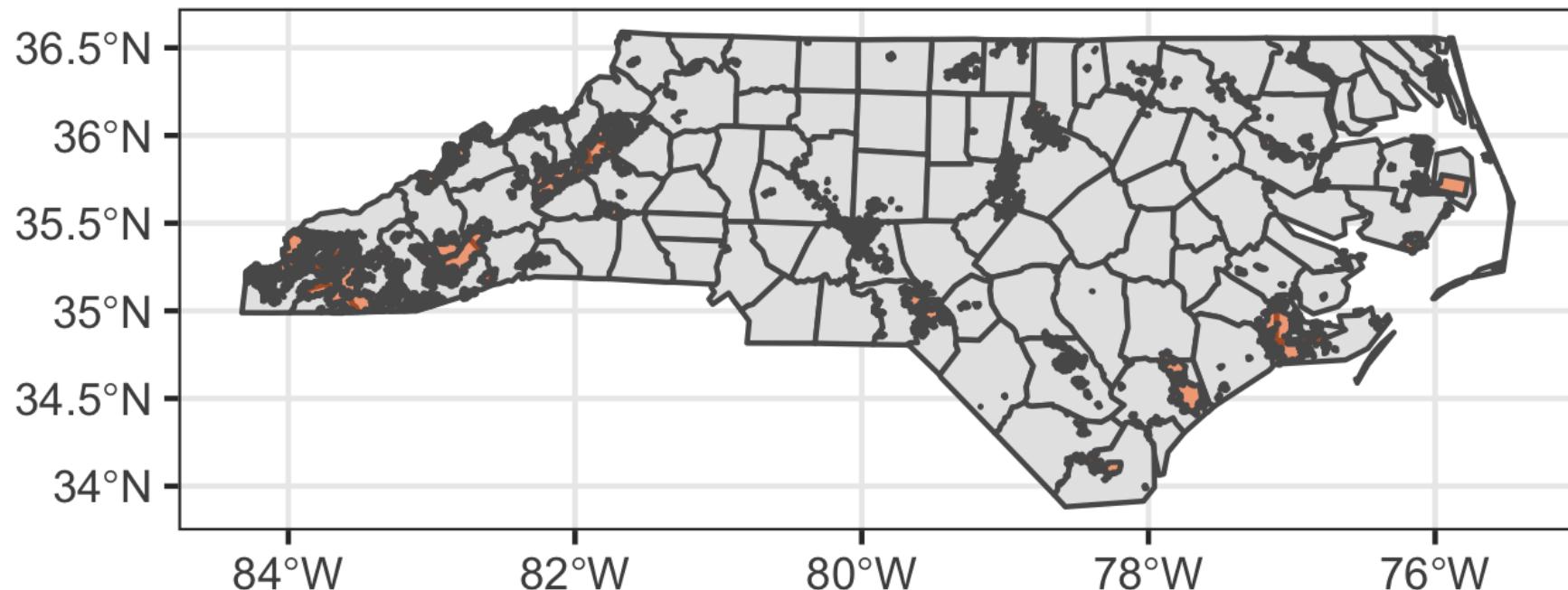
North Carolina gamelands



# Add layers

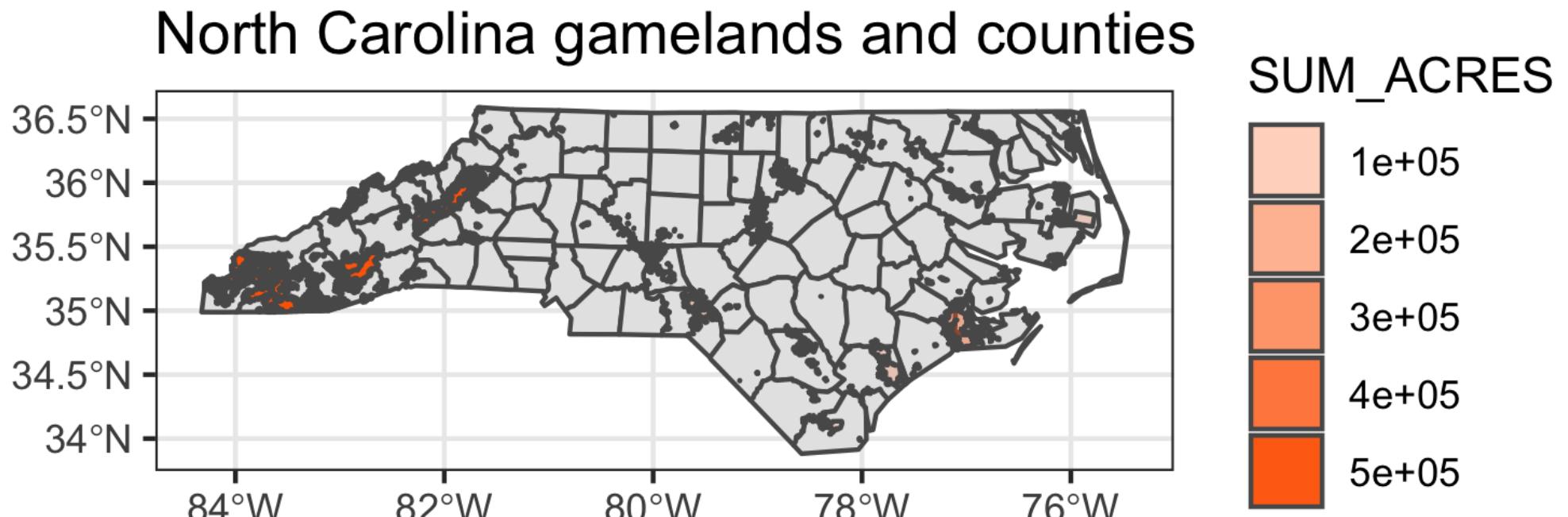
```
ggplot(nc) +  
  geom_sf() +  
  geom_sf(data = nc_game, fill = "#ff6700", alpha = .5) +  
  labs(title = "North Carolina gamelands and counties")
```

North Carolina gamelands and counties



# Add layers and aesthetics

```
ggplot(nc) +  
  geom_sf() +  
  geom_sf(data = nc_game, aes(alpha = SUM_ACRES), fill = "#ff6700") +  
  labs(title = "North Carolina gamelands and counties", fill = "Acres")
```



# Spatial challenges

# Challenge #1

Different types of data exist (raster and vector).



# Challenge #2

The coordinate reference system (CRS) matters.

```
```r
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: 36.58965
epsg (SRID):   4326
proj4string:    +proj=longlat +datum=WGS84 +no_defs
# A tibble: 100 x 2
  NAME               geometry
  <chr>              <MULTIPOLYGON [°]>
1 Ashe    (((-81.47276 36.23436, -81.54084 36.27251, -...
```

```

# Challenge #3

Manipulating spatial data objects is similar, but not identical to manipulating data frames.

Note the core data-wrangling functions from **dplyr** do work.





STA 199

# dplyr + sf



# select

```
nc %>%  
  select(name, regstrd, voted)
```

```
## Simple feature collection with 100 features and 3 fields  
## geometry type: MULTIPOLYGON  
## dimension: XY  
## bbox: xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax: 36.589  
## geographic CRS: NAD27  
## First 10 features:  
##   name regstrd voted      geometry  
## 1 ASHE  19414  8428 MULTIPOLYGON ((((-81.47276 3...  
## 2 ALLEGHANY 7556  4101 MULTIPOLYGON ((((-81.23989 3...  
## 3 SURRY  46666 23660 MULTIPOLYGON ((((-80.45634 3...  
## 4 CURRITUCK 21803  7536 MULTIPOLYGON ((((-76.00897 3...  
## 5 NORTHAMPTON 13891  6196 MULTIPOLYGON ((((-77.21767 3...  
## 6 HERTFORD 14945  6955 MULTIPOLYGON ((((-76.74506 3...  
## 7 CAMDEN 8128  3472 MULTIPOLYGON ((((-76.00897 3...  
## 8 GATES 8294  3195 MULTIPOLYGON ((((-76.56251 3...
```



# filter

```
nc %>%  
  filter(regstrd > 100000)
```

```
## Simple feature collection with 20 features and 6 fields  
## geometry type:  MULTIPOLYGON  
## dimension:      XY  
## bbox:           xmin: -82.88111 ymin: 33.88199 xmax: -77.10377 ymax: 36.257  
## geographic CRS: NAD27  
## First 10 features:  
##          name regstrd voted mailed rejectd ml_rqst  
## 1 FORSYTH    270818 134770  35664       75   67472  
## 2 GUILFORD   381797 190530  42825      785   76615  
## 3 ALAMANCE   110127  52434  12788       19   22091  
## 4 ORANGE     111765  64016  22859       55   36560  
## 5 DURHAM     243045 138264  41767      348   70046  
## 6 WAKE        791821 421180 138483      781  244538  
## 7 IREDELL    130013  67128  14775        2   22390  
## 8 DAVIDSON   112872  54940  10012       87   16249
```



# summarize

```
nc %>%
  summarize(mean_registered = mean(regstrd),
            mean_voted = median(voted))
```

```
## Simple feature collection with 1 feature and 2 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:             xmin: -84.32385  ymin: 33.88199  xmax: -75.45698  ymax: 36.589
## geographic CRS: NAD27
##   mean_registered  mean_voted
## 1       73183.36      16217 MULTIPOLYGON (((-77.96073 3...
```



# Geometries are "sticky"

The geometry is kept until it is deliberately dropped using `st_drop_geometry`.



# Geometries are "sticky"

The geometry is kept until it is deliberately dropped using **st\_drop\_geometry**.

```
nc %>%
  select(name, regstrd) %>%
  filter(regstrd > 100000) %>%
  st_drop_geometry()
```

```
##           name regstrd
## 1      FORSYTH    270818
## 2     GUILFORD    381797
## 3     ALAMANCE    110127
## 4      ORANGE     111765
## 5     DURHAM      243045
## 6       WAKE      791821
## 7    IREDELL     130013
## 8   DAVIDSON     112872
## 9       PITT      122925
## 10    CATALINA    100000
```



# References

- North Carolina Early Voting Statistics
- Simple Features for R vignette
- mapview vignette
- Coordinate Reference Systems in R
- Geocomputation with R

