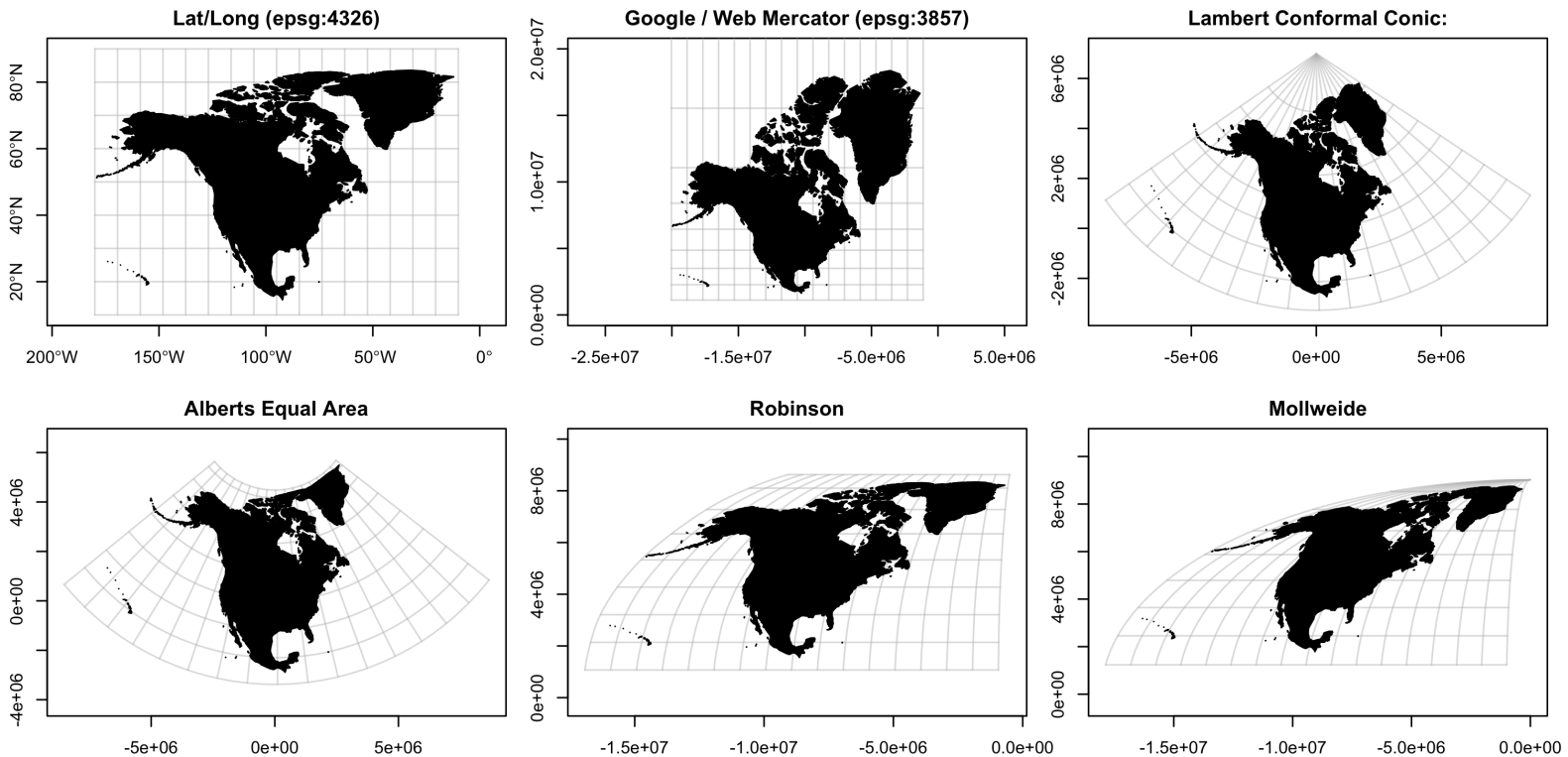
Spatial data & GIS tools

Lecture 21

Dr. Colin Rundel

Geospatial stuff is hard

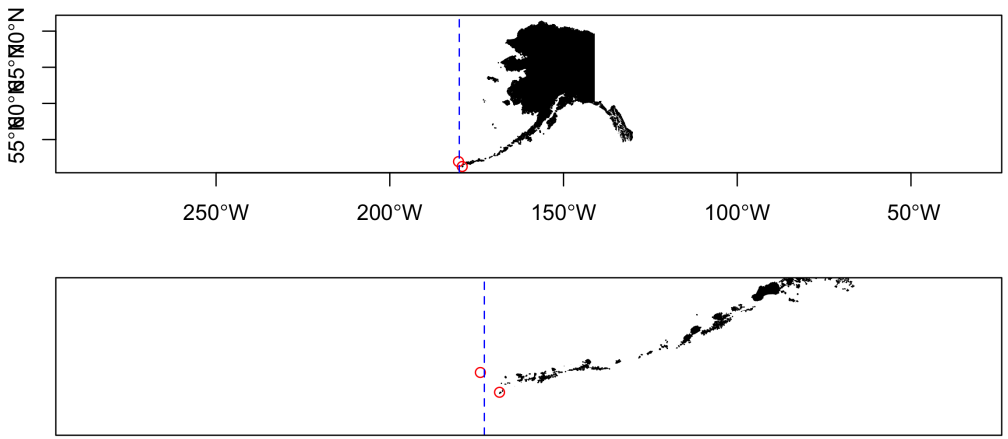
Projections



EPSG is a public registry of coordinate reference systems and related data (name comes from the European Petroleum Survey Group)

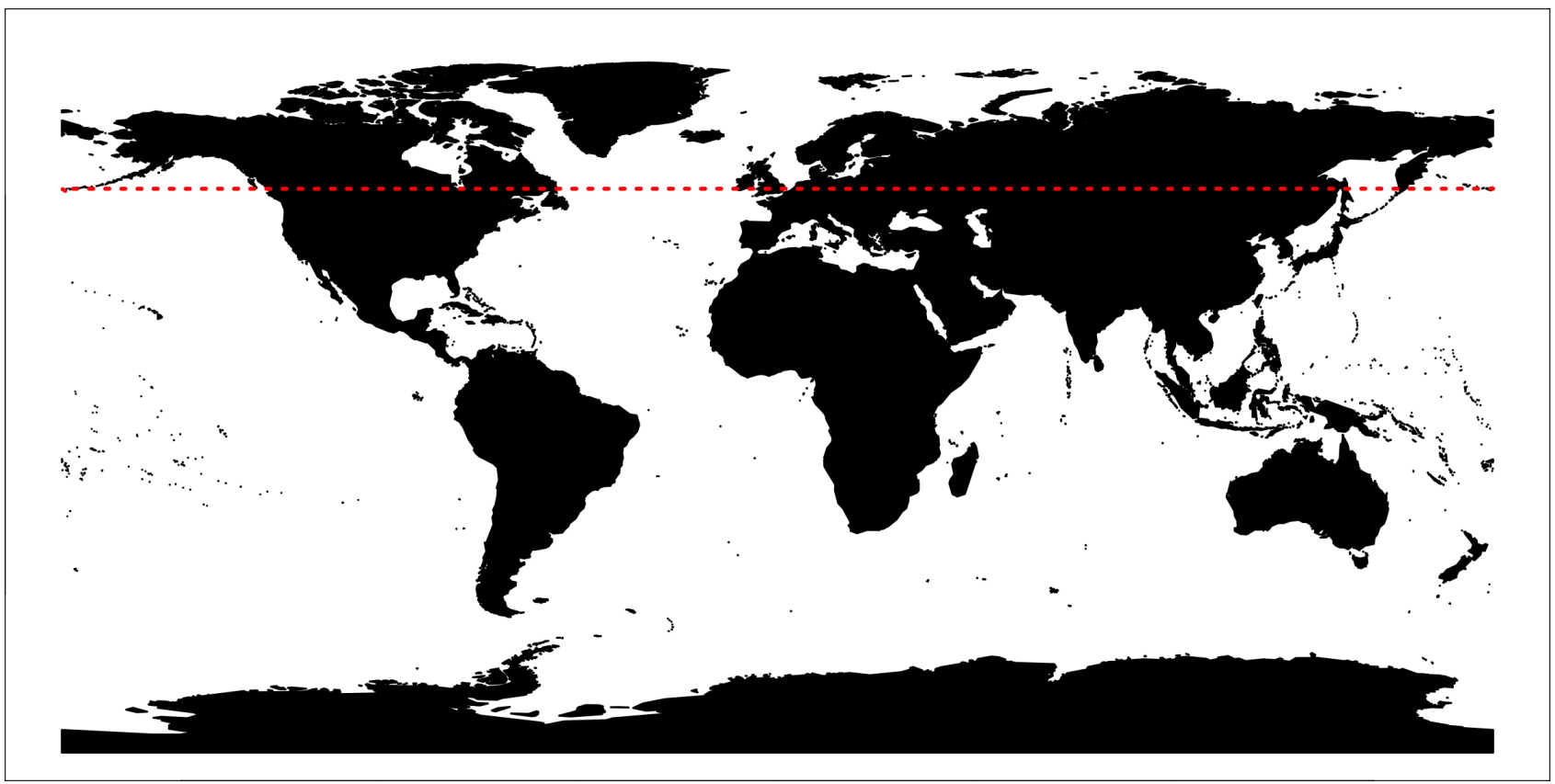
Dateline

How long is the flight between the Western most and the Eastern most points in the US?

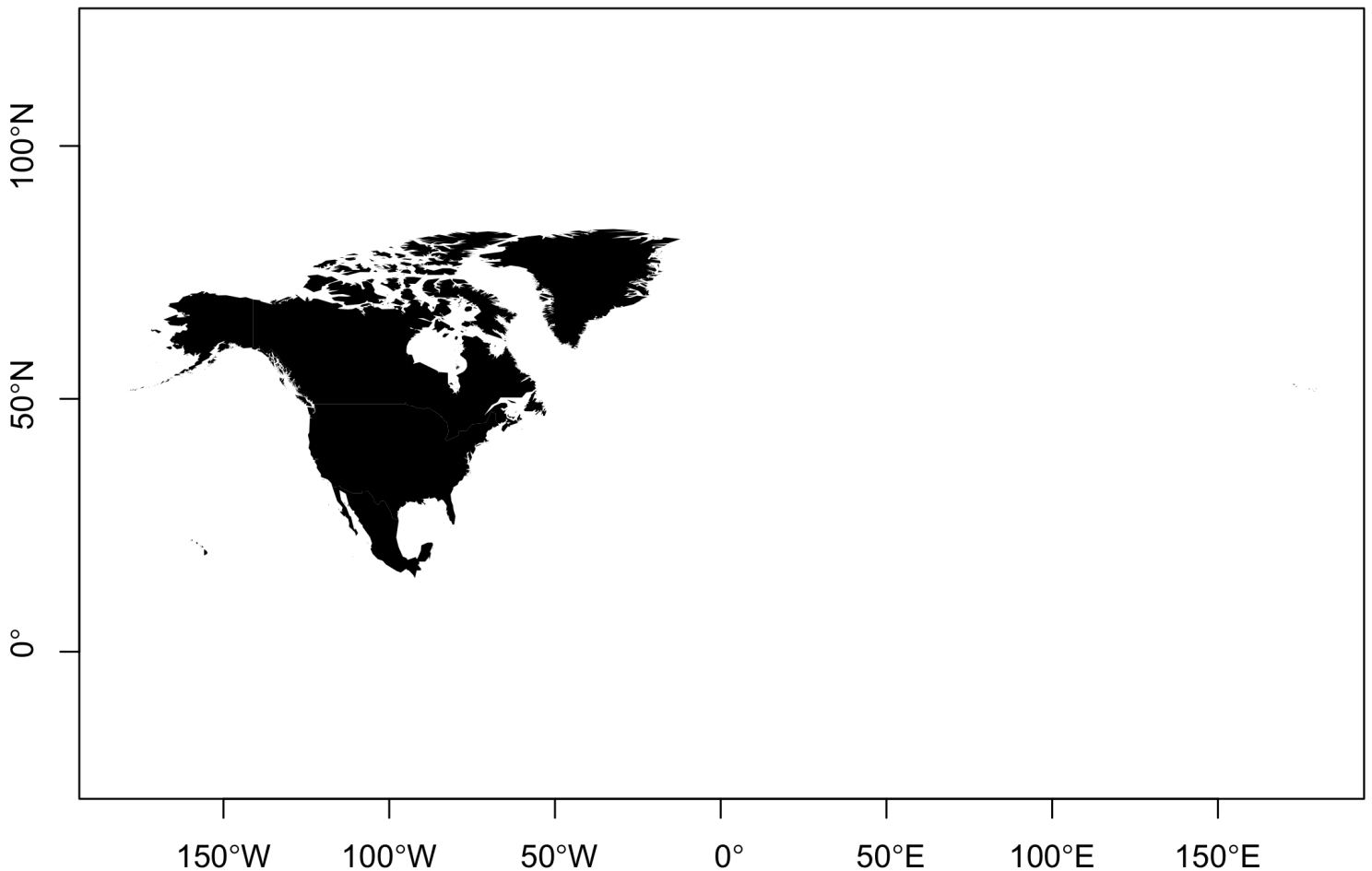


Great circle distance

```
par(mar=c(0,0,0,0))
ak1 = c(179.776, 51.952)
ak2 = c(-179.146, 51.273)
inter = geosphere::gcIntermediate(ak1, ak2, n=50, addStartEnd=TRUE)
plot(st_geometry(world), col="black", ylim=c(-90,90), axes=TRUE)
lines(inter,col='red',lwd=2,lty=3)
```

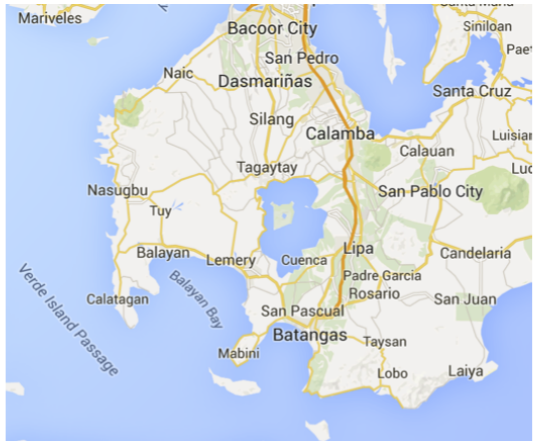


Plotting is hard



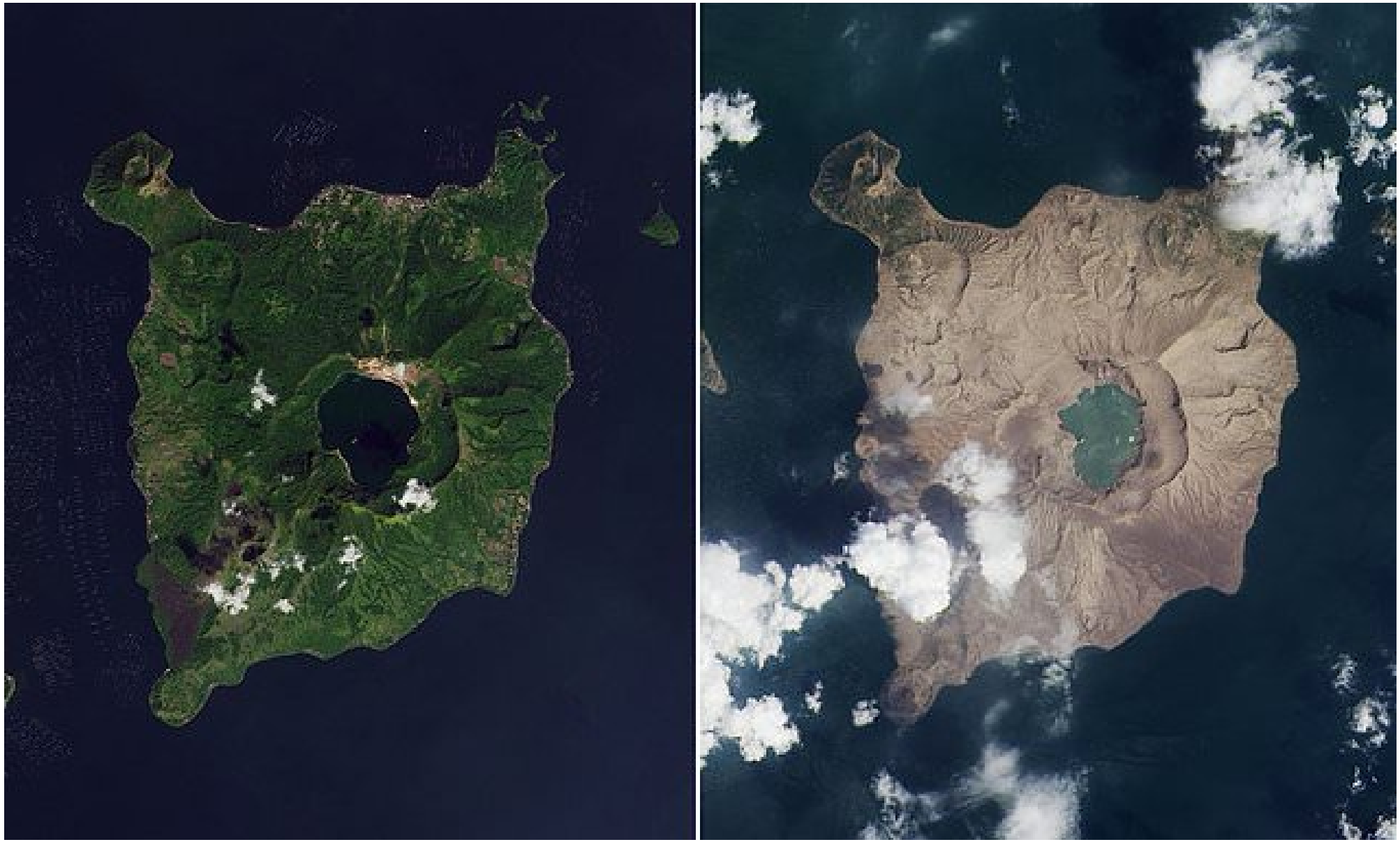
Relationships


Kalusunan
South
hina Sea
Manila
Philippines
Spratly
Islands
Kabisayaan
Palawan
Negros
Sulu Sea
Mindanao
Davao
SABAH


Mariveles
Bacoor City
Siniloan
San Pedro
Naic
Dasmariñas
Santa Cruz
Silang
Calamba
Calauan
Tagaytay
Nasugbu
San Pablo City
Tuy
Balayan
Lemery
Cuenca
Lipa
Candelaria
Padre Garcia
Rosario
Calatagan
Balayan Bay
San Juan
San Pascual
Batangas
Taysan
Verde Island Passage
Mabini
Lobo
Laiya


Tanauan
Balete
Taal Volcano Island
Lawa ng Taal
Mataas
Na Kahoy
Agoncillo
San Nicolas
Cuenca
Taal
Santa Teresita
Southern Tagalog Arterial Road


Taal Volcano Island



Source

Geospatial Data and R

Packages for geospatial data in R

R has a rich package ecosystem for read/writing, manipulating, and analyzing geospatial data.

- `sp` - core classes for handling spatial data, additional utility functions - **Deprecated**
- `rgdal` - R interface to `gdal` (Geospatial Data Abstraction Library) for reading and writing spatial data - **Deprecated**
- `rgeos` - R interface to `geos` (Geometry Engine Open Source) library for querying and manipulating spatial data. Reading and writing WKT. - **Deprecated**
- `sf` - Combines the functionality of `sp`, `rgdal`, and `rgeos` into a single package based on tidy simple features.
- `raster` - classes and tools for handling spatial raster data.
- `stars` - Reading, manipulating, writing and plotting spatiotemporal arrays (rasters)

See more - Spatial task view

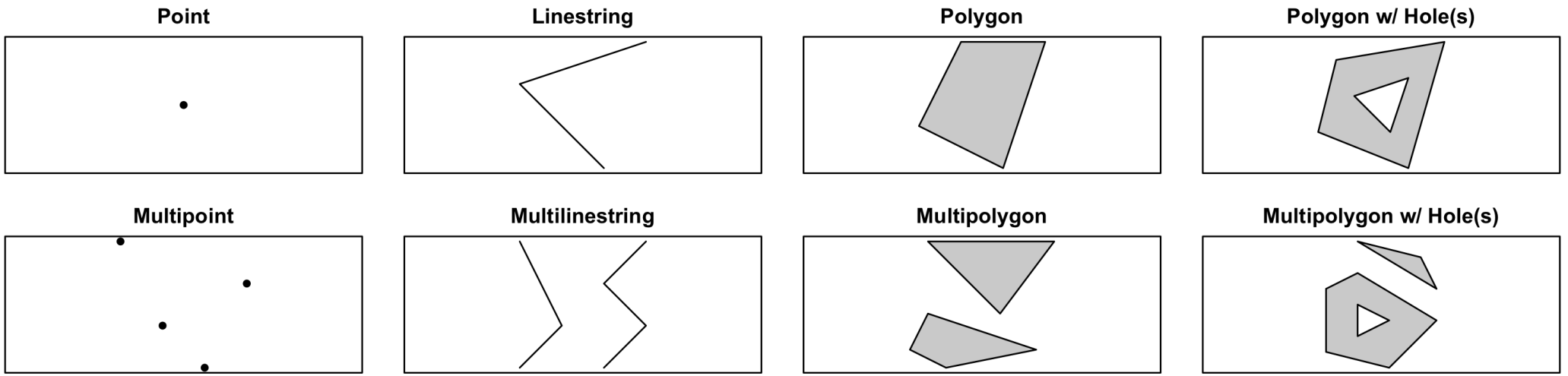
Installing `sf`

This is the hardest part of using the `sf` package, difficulty comes from is dependence on several external libraries (`geos`, `gdal`, and `proj`).

- *Windows* - installing from source works when Rtools is installed (system requirements are downloaded from rwinlib)
- *MacOS* - install dependencies via homebrew: `gdal`, `geos`, `proj`, `udunits`.
- *Linux* - Install development pacakages for GDAL (>= 2.0.0), GEOS (>= 3.3.0), Proj4 (>= 4.8.0), udunits2 from your package manager of choice.

More specific details are included in the repo README on github.

Simple Features



Reading, writing, and converting simple features

- `st_read` / `st_write` - Shapefile, GeoJSON, KML, ...
- `read_sf` / `write_sf` - Same, suports tibbles ...
- `st_as_sfc` / `st_as_wkt` - WKT
- `st_as_sfc` / `st_as_binary` - WKB
- `st_as_sfc` / `as(x, "Spatial")` - sp

See sf vignette #2 - Reading, Writing and Converting Simple Features.

Shapefiles

```
1 fs::dir_info("data/gis/nc_counties/") %>% select(path:size)
```

```
# A tibble: 4 × 3
  path                                  type     size
  <fs::path>                            <fct> <fs::b>
1 …a/gis/nc_counties/nc_counties.dbf    file      41K
2 …a/gis/nc_counties/nc_counties.prj    file      165
3 …a/gis/nc_counties/nc_counties.shp    file    1.41M
4 …a/gis/nc_counties/nc_counties.shx    file      900
```

NC Counties

```
1 (st_read("data/gis/nc_counties/", quiet=FALSE))
```

```
Reading layer `nc_counties' from data source
  `/Users/rundel/Desktop/Sta523-Fa22/website/static/slides/data/gis/nc_counties'
  using driver `ESRI Shapefile'
Simple feature collection with 100 features and 8 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: -84.32186 ymin: 33.84175 xmax: -75.46003 ymax: 36.58815
Geodetic CRS:  NAD83

Simple feature collection with 100 features and 8 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: -84.32186 ymin: 33.84175 xmax: -75.46003 ymax: 36.58815
Geodetic CRS:  NAD83
First 10 features:
         AREA PERIMETER COUNTYP010 STATE
1  0.11175964  1.610396       1994    NC
2  0.06159483  1.354829       1996    NC
3  0.14023009  1.769388       1998    NC
4  0.08912401  1.425249       1999    NC
5  0.06865730  4.428217       2000    NC
6  0.11859434  1.404309       2001    NC
7  0.06259671  2.106357       2002    NC
8  0.11542955  1.462524       2003    NC
9  0.14328609  2.397293       2004    NC
```

sf tibbles

```
1  (nc = read_sf("data/gis/nc_counties/"))
```

```
Simple feature collection with 100 features and 8 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: -84.32186 ymin: 33.84175 xmax: -75.46003 ymax: 36.58815
Geodetic CRS:  NAD83
# A tibble: 100 × 9
     AREA PERIMETER COUNTYP010 STATE COUNTY  FIPS
    <dbl>     <dbl>      <dbl> <chr> <chr>   <chr>
 1 0.112       1.61       1994 NC    Ashe C… 37009
 2 0.0616      1.35       1996 NC    Allegh… 37005
 3 0.140       1.77       1998 NC    Surry … 37171
 4 0.0891      1.43       1999 NC    Gates … 37073
 5 0.0687      4.43       2000 NC    Currit… 37053
 6 0.119       1.40       2001 NC    Stokes… 37169
 7 0.0626      2.11       2002 NC    Camden… 37029
 8 0.115       1.46       2003 NC    Warren… 37185
 9 0.143       2.40       2004 NC    Northa… 37131
```

sf classes

```
str(nc, max.level=1)
```

```
sf [100 × 9] (S3: sf/tbl_df/tbl/data.frame)
 - attr(*, "sf_column")= chr "geometry"
 - attr(*, "agr")= Factor w/ 3 levels "constant","aggregate",..: NA NA NA NA NA
NA NA NA
  ..- attr(*, "names")= chr [1:8] "AREA" "PERIMETER" "COUNTYP010" "STATE" ...
```

```
class(nc)
```

```
[1] "sf"         "tbl_df"     "tbl"
[4] "data.frame"
```

```
class(nc$geometry)
```

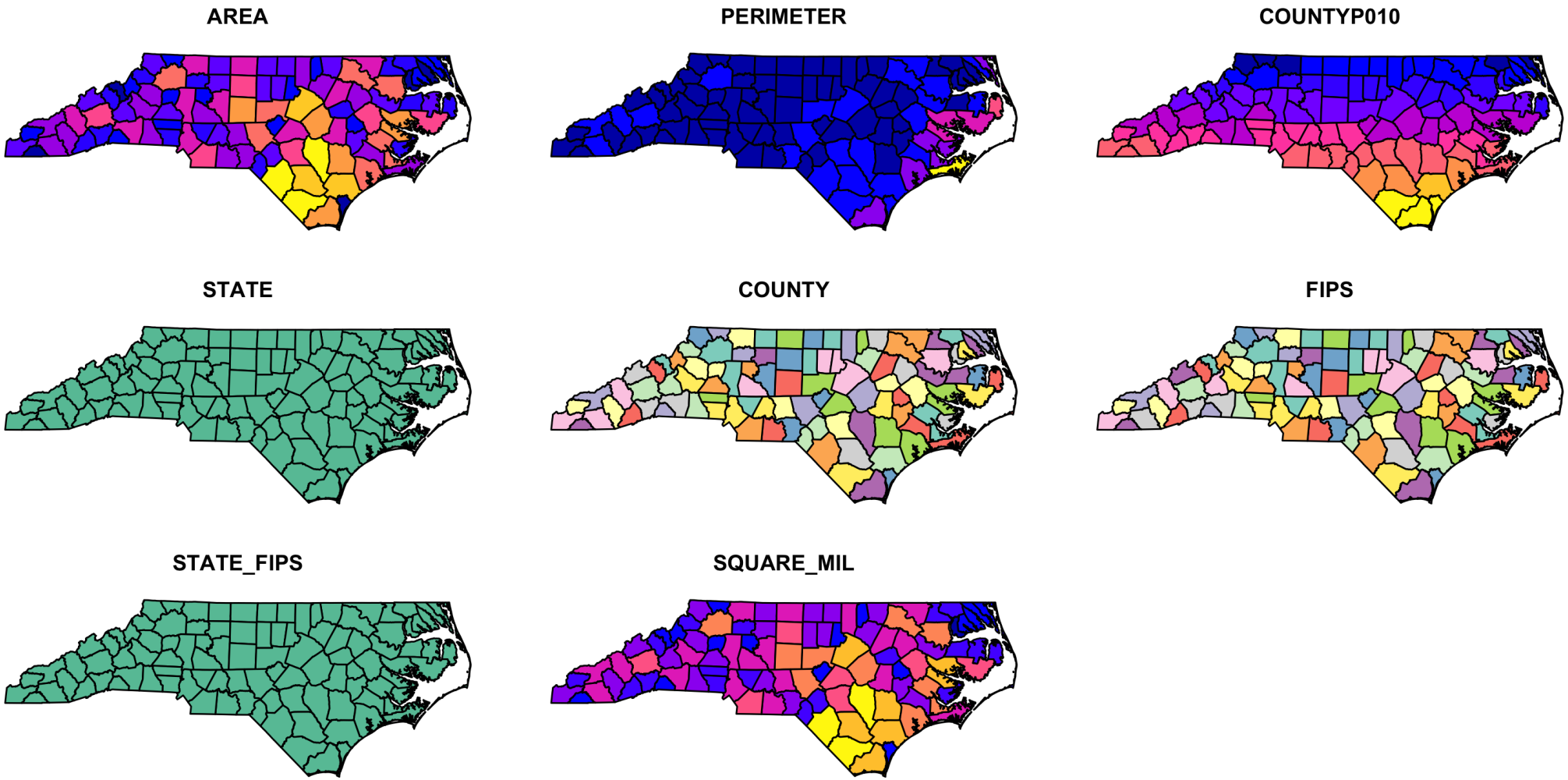
```
[1] "sfc_MULTIPOLYGON" "sfc"
```

```
class(nc$geometry[[1]])
```

```
[1] "XY"           "MULTIPOLYGON" "sfg"
```

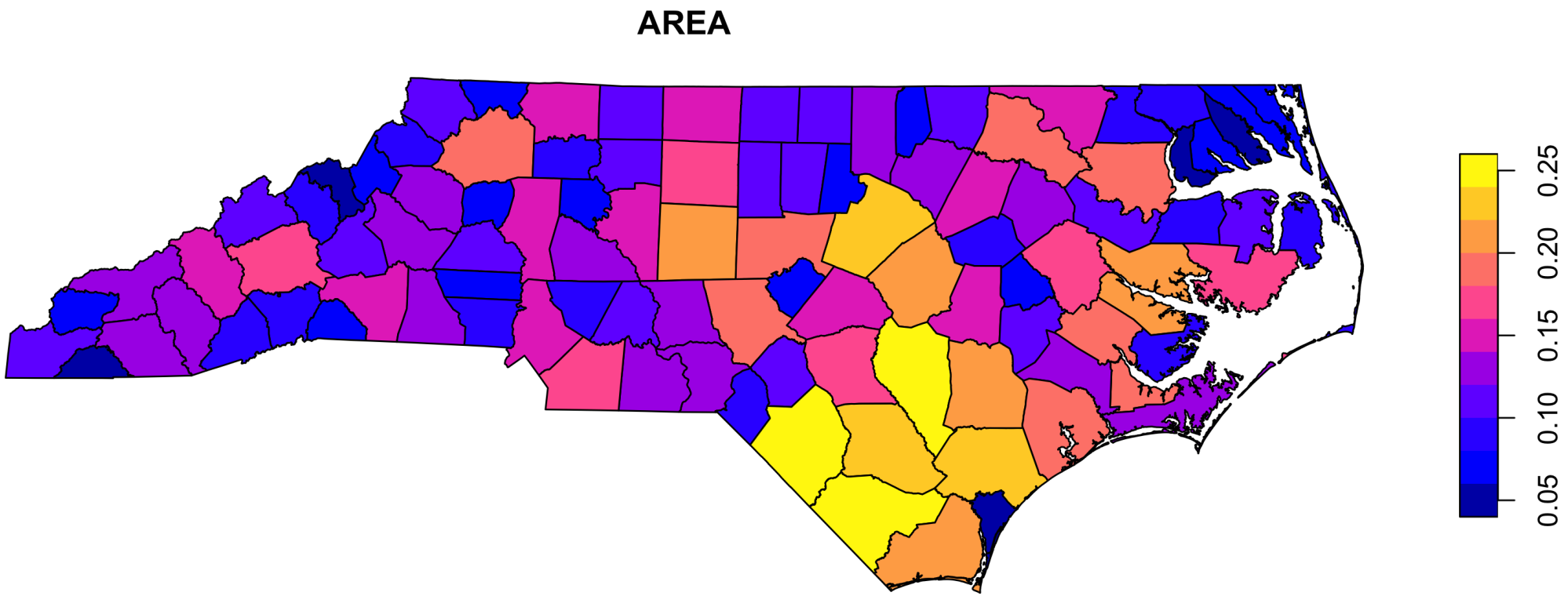
Plotting

```
1 plot(nc)
```



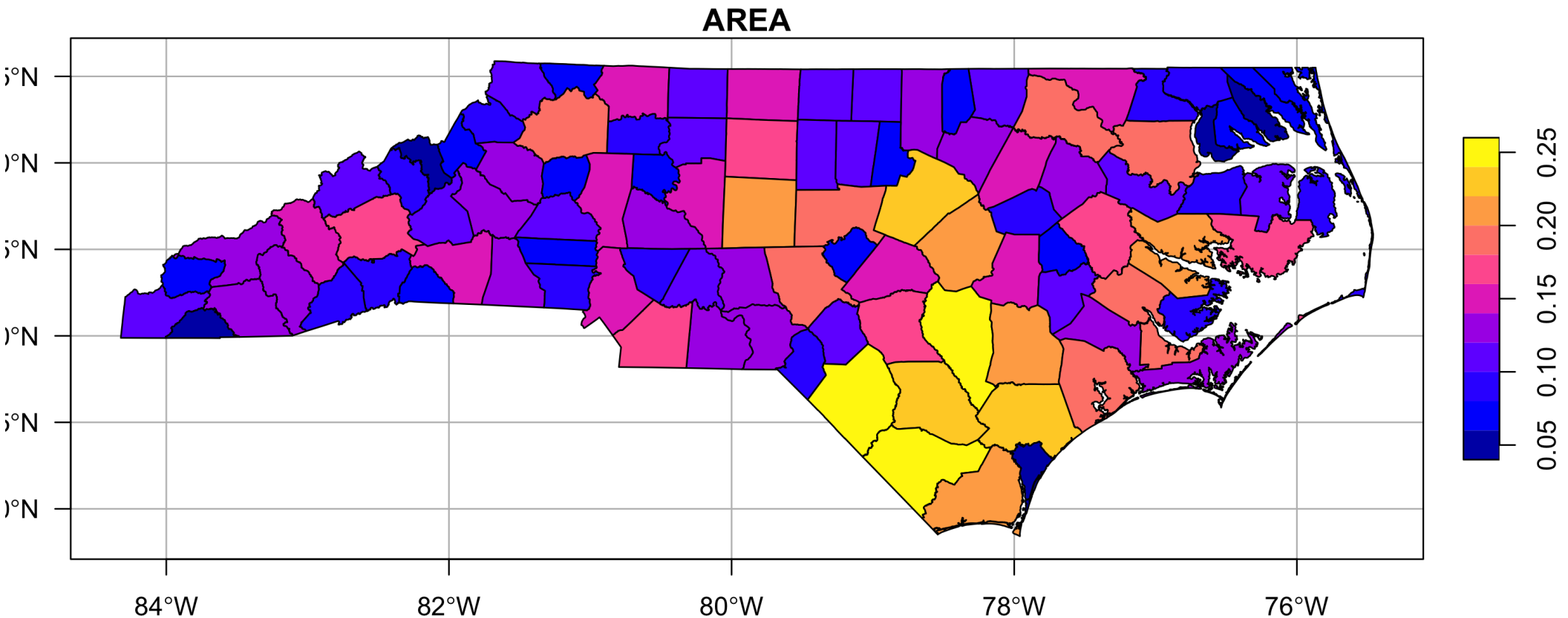
More Plotting

```
1  plot(nc["AREA"])
```



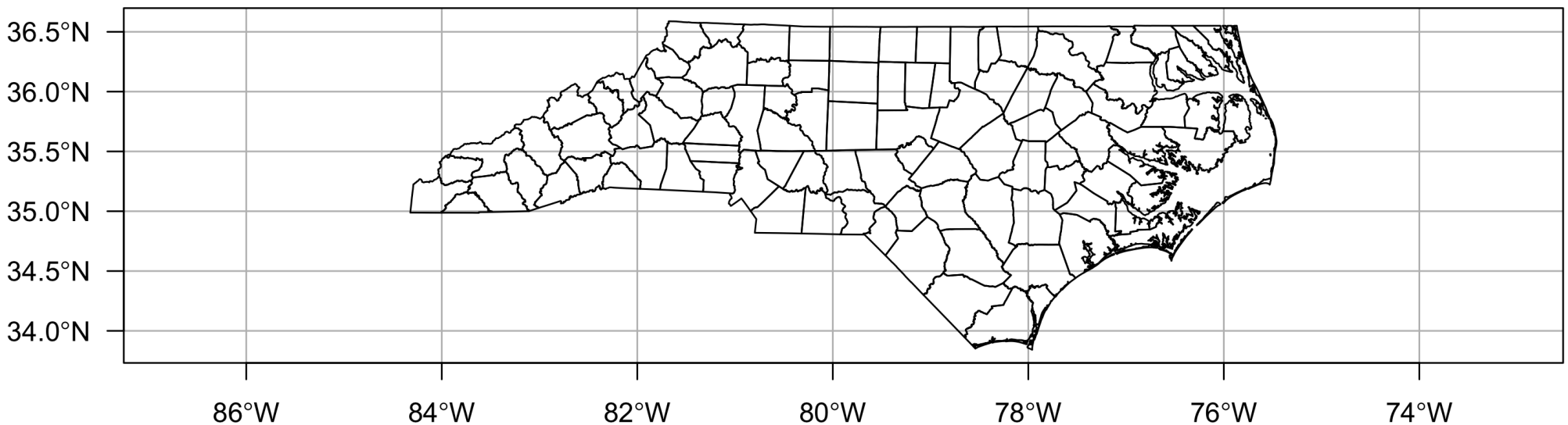
Graticules

```
1 plot(nc["AREA"], graticule=TRUE, axes=TRUE, las=1)
```



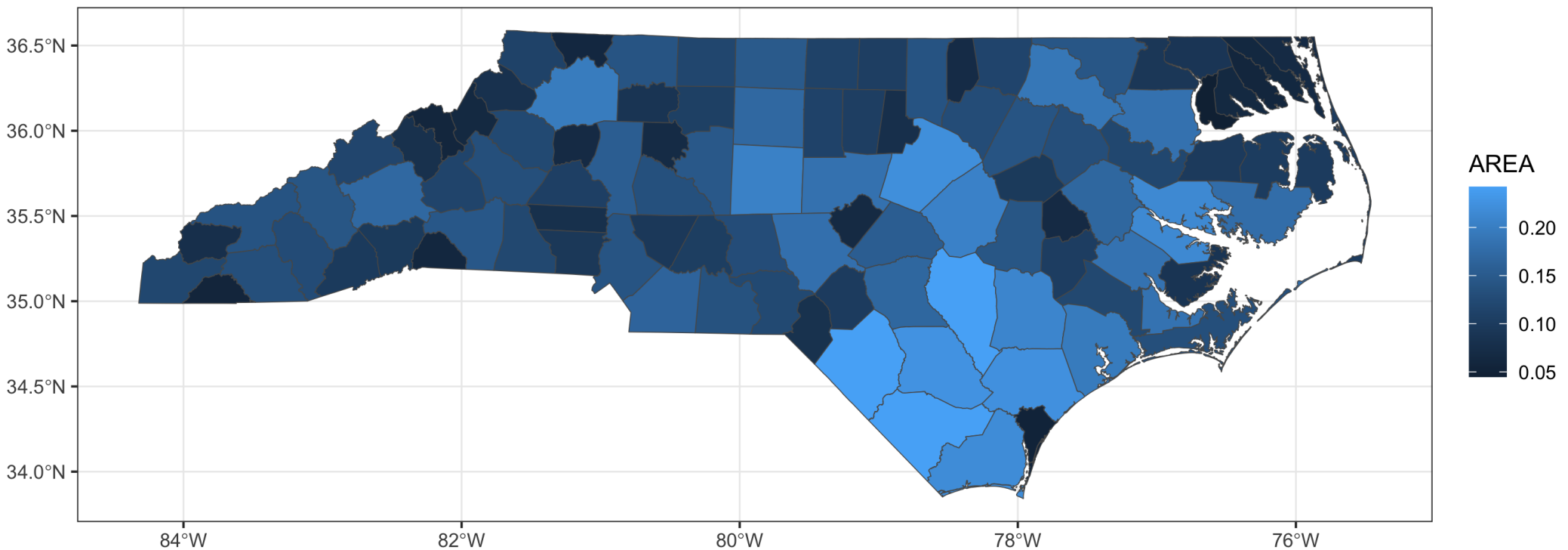
Geometries

```
1  plot(st_geometry(nc), graticule=TRUE, axes=TRUE, las=1)
```



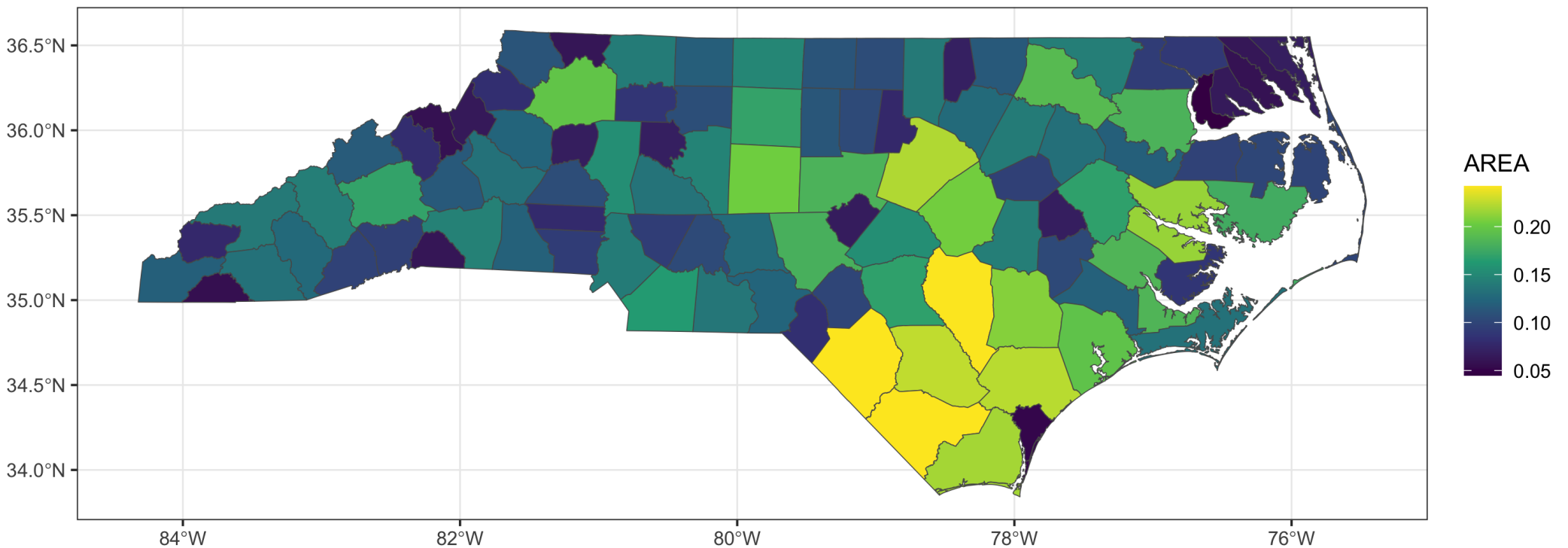
ggplot2

```
1  ggplot(nc, aes(fill=AREA)) +
2    geom_sf()
```



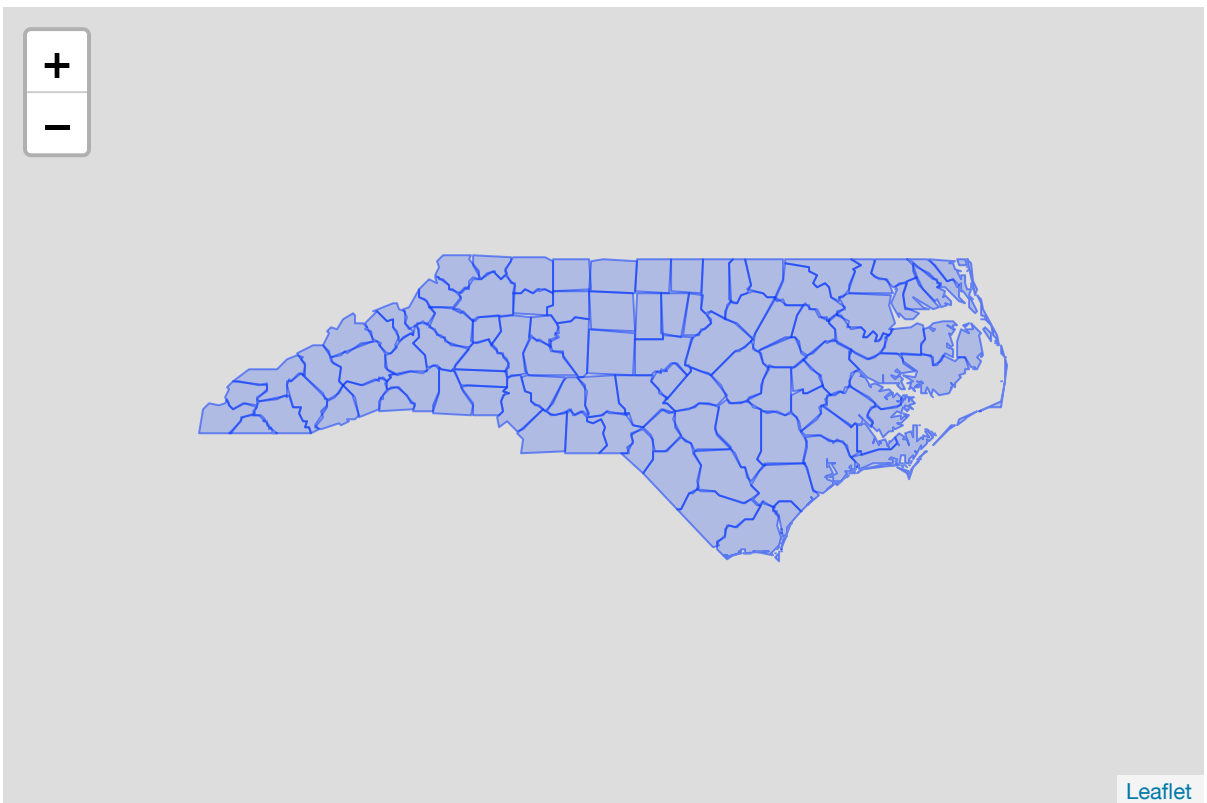
ggplot2 + palettes

```
1  ggplot(nc, aes(fill=AREA)) +
2    geom_sf() +
3    scale_fill_viridis_c()
```



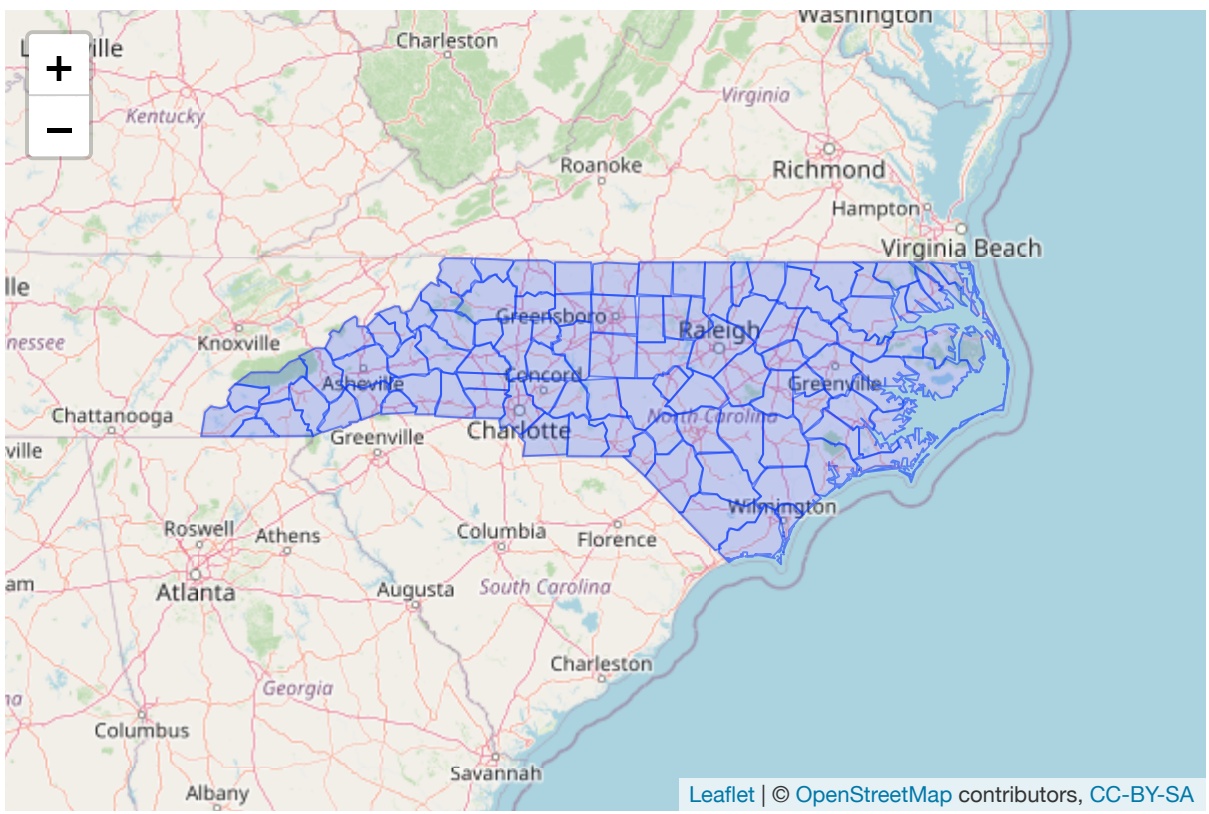
leaflet

```
st_transform(nc, "+proj=longlat +datum=WGS84") %>%
leaflet::leaflet(width = 600, height = 400) %>%
  leaflet::addPolygons(
    weight = 1,
    popup = ~COUNTY,
    highlightOptions = leaflet::highlightOptions(color = "red", weight = 2, bringToFront = TRUE)
  )
```



leaflet + tiles

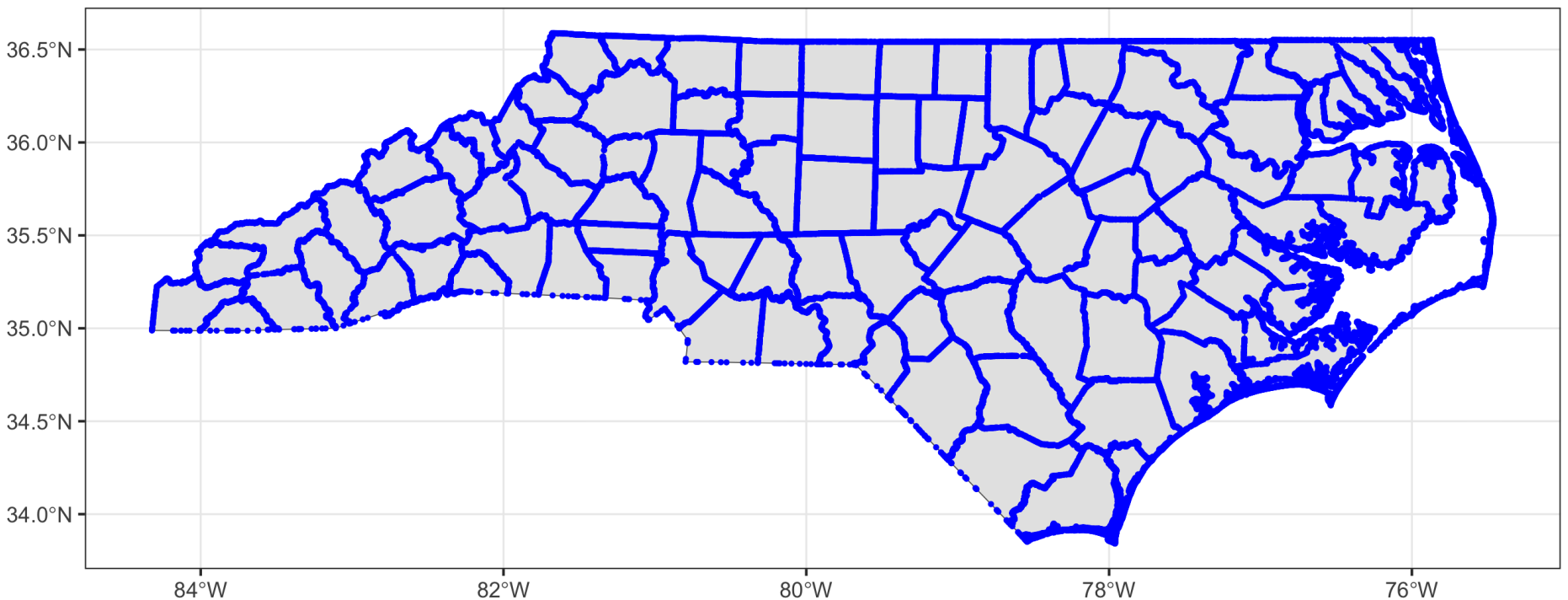
```
st_transform(nc, "+proj=longlat +datum=WGS84") %>%
leaflet::leaflet(width = 600, height = 400) %>%
  leaflet::addPolygons(
    weight = 1,
    popup = ~COUNTY,
    highlightOptions = leaflet::highlightOptions(color = "red", weight = 2, bringToFront = TRUE)
  ) %>%
  leaflet::addTiles()
```



GIS in R

Geometry casting

```
nc_pts = st_cast(nc, "MULTIPOINT")
ggplot() +
  geom_sf(data=nc) +
  geom_sf(data=nc_pts, size=0.5, color="blue")
```

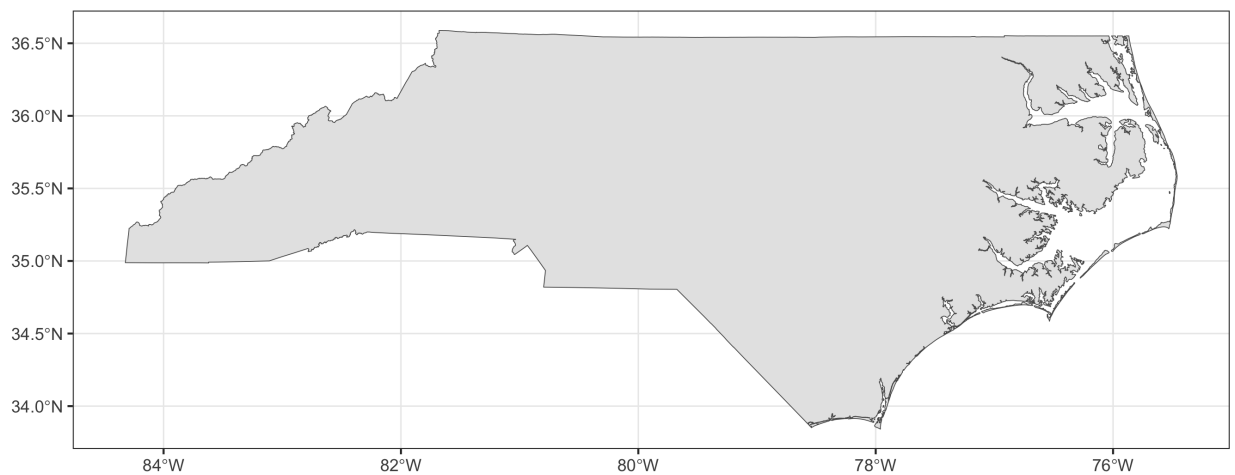


Joining

```
1  (nc_state = st_union(nc))
```

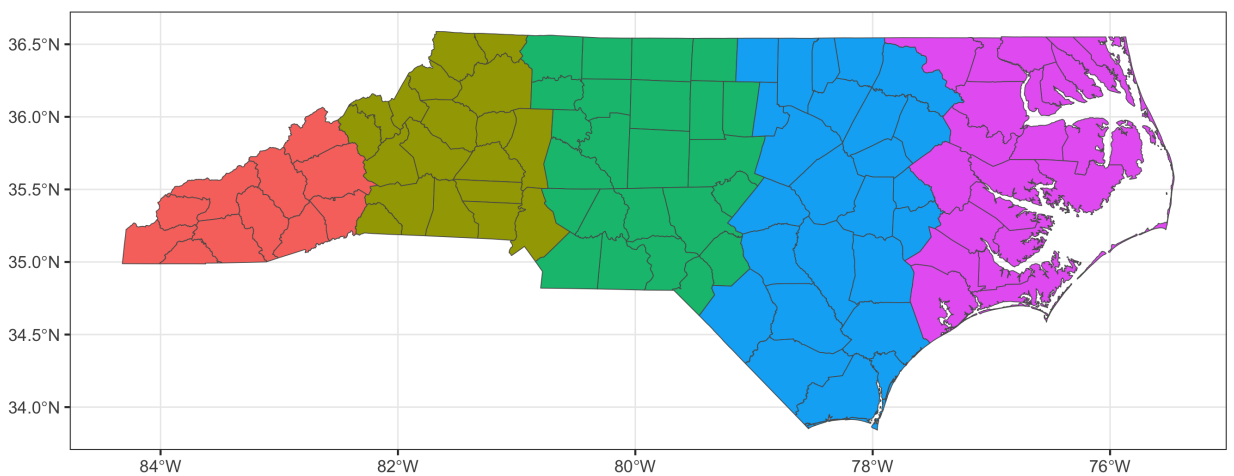
```
Geometry set for 1 feature
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: -84.32186 ymin: 33.84175 xmax: -75.46003 ymax: 36.58815
Geodetic CRS:  NAD83
```

```
1  ggplot() + geom_sf(data=nc_state)
```



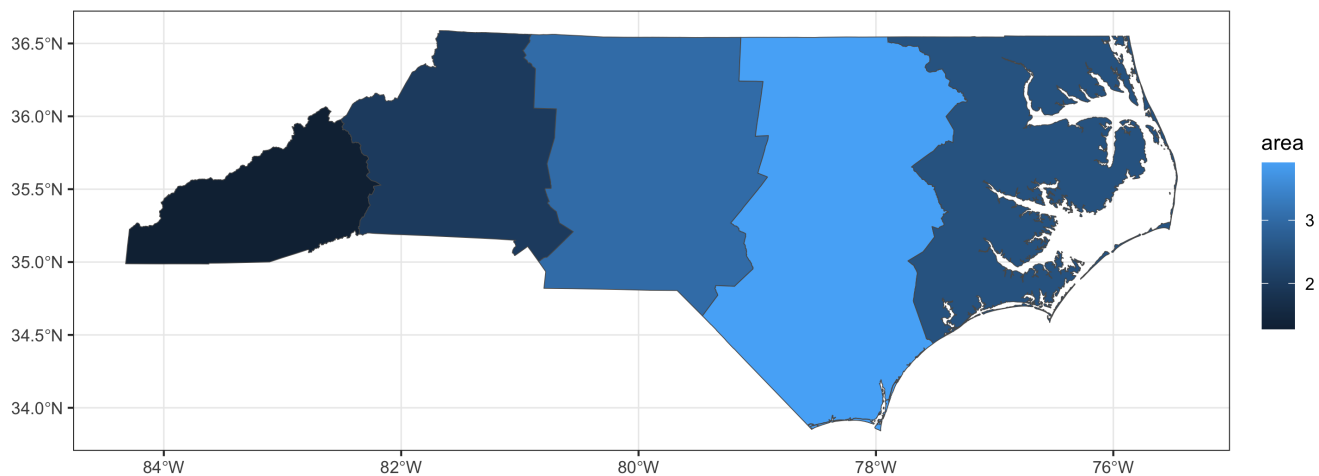
sf & dplyr

```
nc_cut = nc %>%
  mutate(
    ctr_x = st_centroid(nc) %>% st_coordinates() %>% .[,1],
    region = cut(ctr_x, breaks = 5)
  )

ggplot(nc_cut) +
  geom_sf(aes(fill=region)) +
  guides(fill = "none")
```



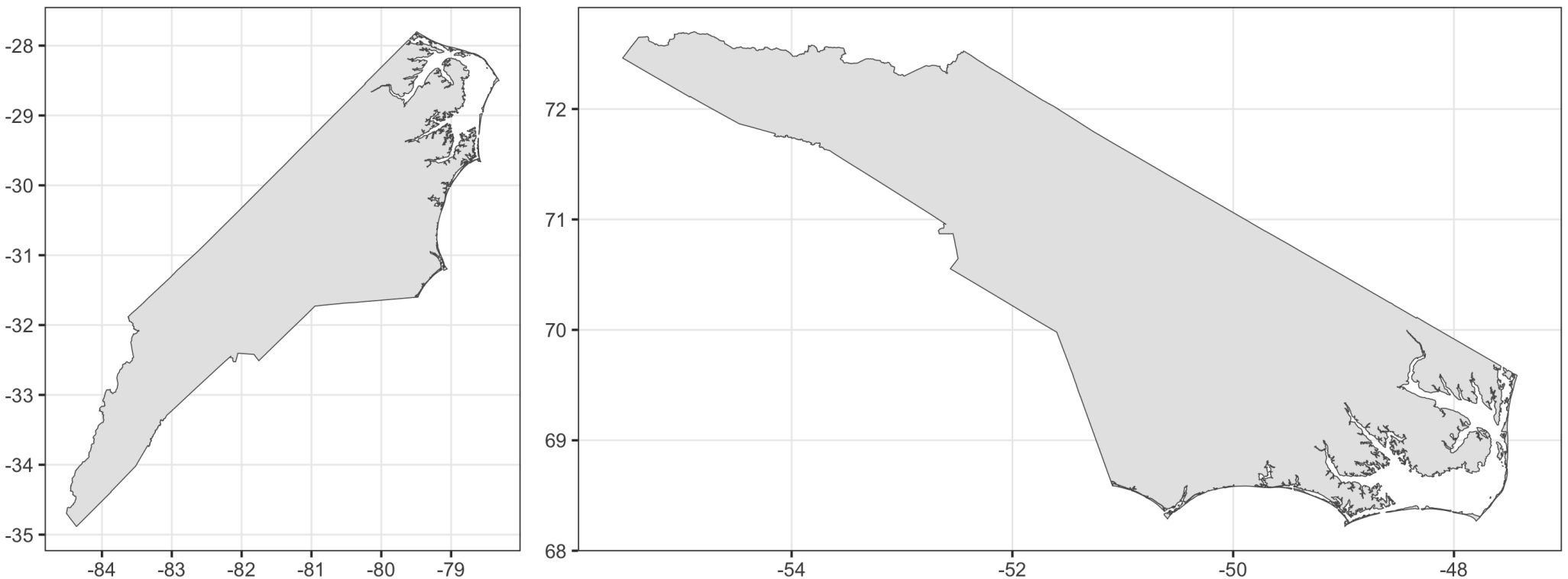
sf & dplyr (cont.)

```
nc_cut2 = nc_cut %>%
  group_by(region) %>%
  summarize(
    area = sum(AREA)
  )

ggplot() + geom_sf(data=nc_cut2, aes(fill=area))
```



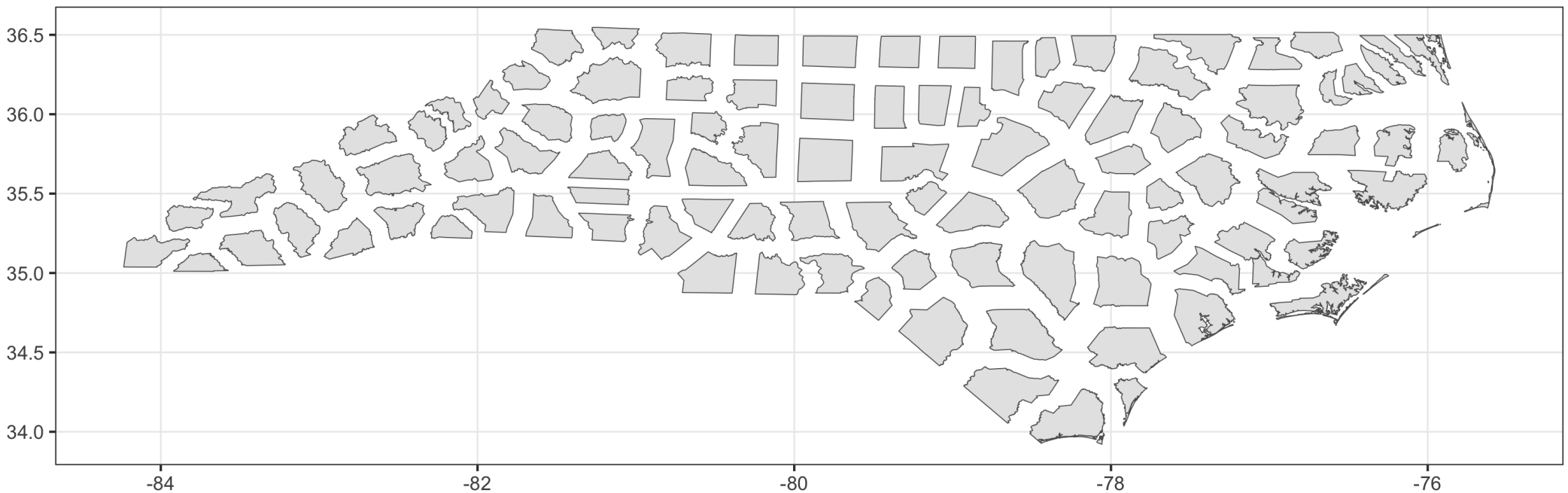
Affine Transformations

```
1 rotate = function(a) matrix(c(cos(a), sin(a), -sin(a), cos(a)), 2, 2)
2
3 (ggplot() + geom_sf(data=(nc_state) * rotate(-pi/4))) +
4 (ggplot() + geom_sf(data=(nc_state) * rotate(pi/6)))
```



Scaling + Translations

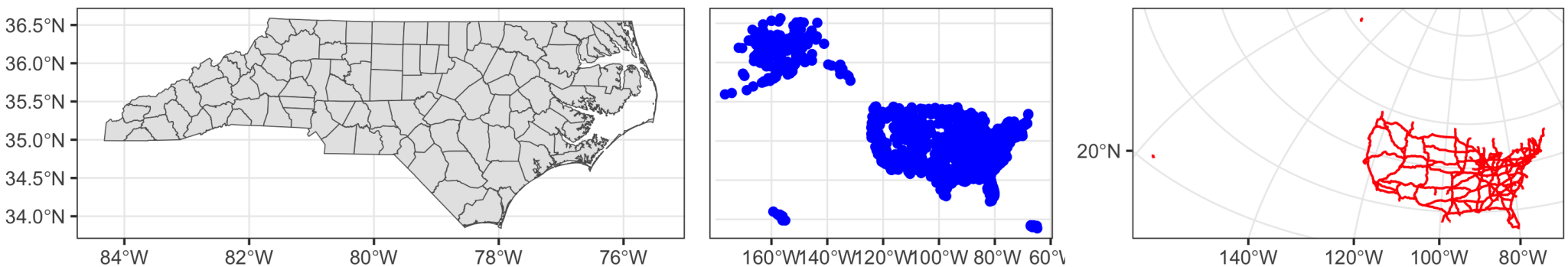
```
1 ctrd = st_centroid(st_geometry(nc))
2 nc_scaled = (st_geometry(nc) - ctrd) * 0.66 + ctrd
3
4 ggplot() + geom_sf(data=nc_scaled)
```



Some other data

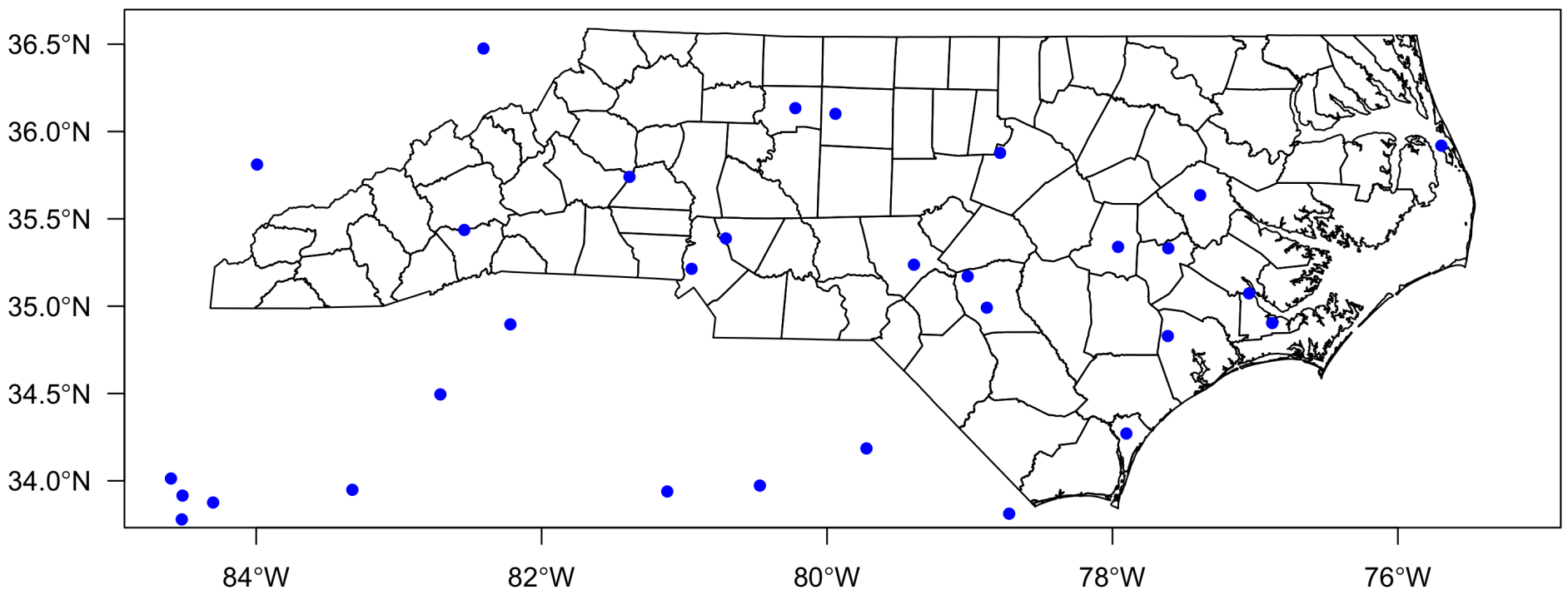
```
air = read_sf("data/gis/airports/", quiet=TRUE)
hwy = read_sf("data/gis/us_interstates/", quiet=TRUE)
```

```
(ggplot(nc) + geom_sf()) +
(ggplot(air) + geom_sf(color = "blue")) +
(ggplot(hwy) + geom_sf(color = "red"))
```



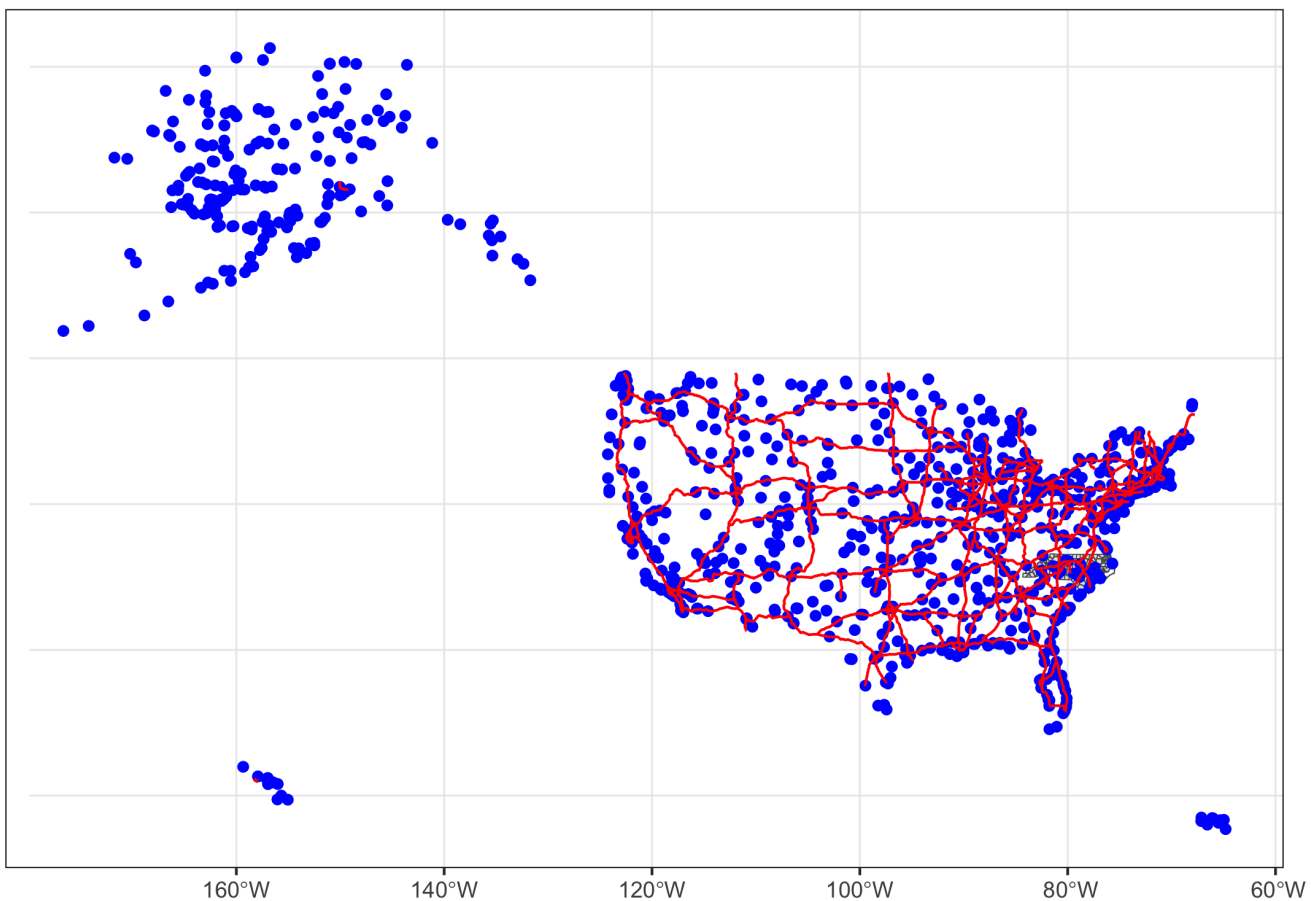
Overlays?

```
1 plot(st_geometry(nc),  axes=TRUE, las=1)
2 plot(st_geometry(air), axes=TRUE, pch=16, col="blue", main="air", add=TRUE)
3 plot(st_geometry(hwy), axes=TRUE, col="red", add=TRUE)
```



Overlays? (ggplot)

```
ggplot() +
  geom_sf(data=nc) +
  geom_sf(data=air, color="blue") +
  geom_sf(data=hwy, color="red")
```



Projections

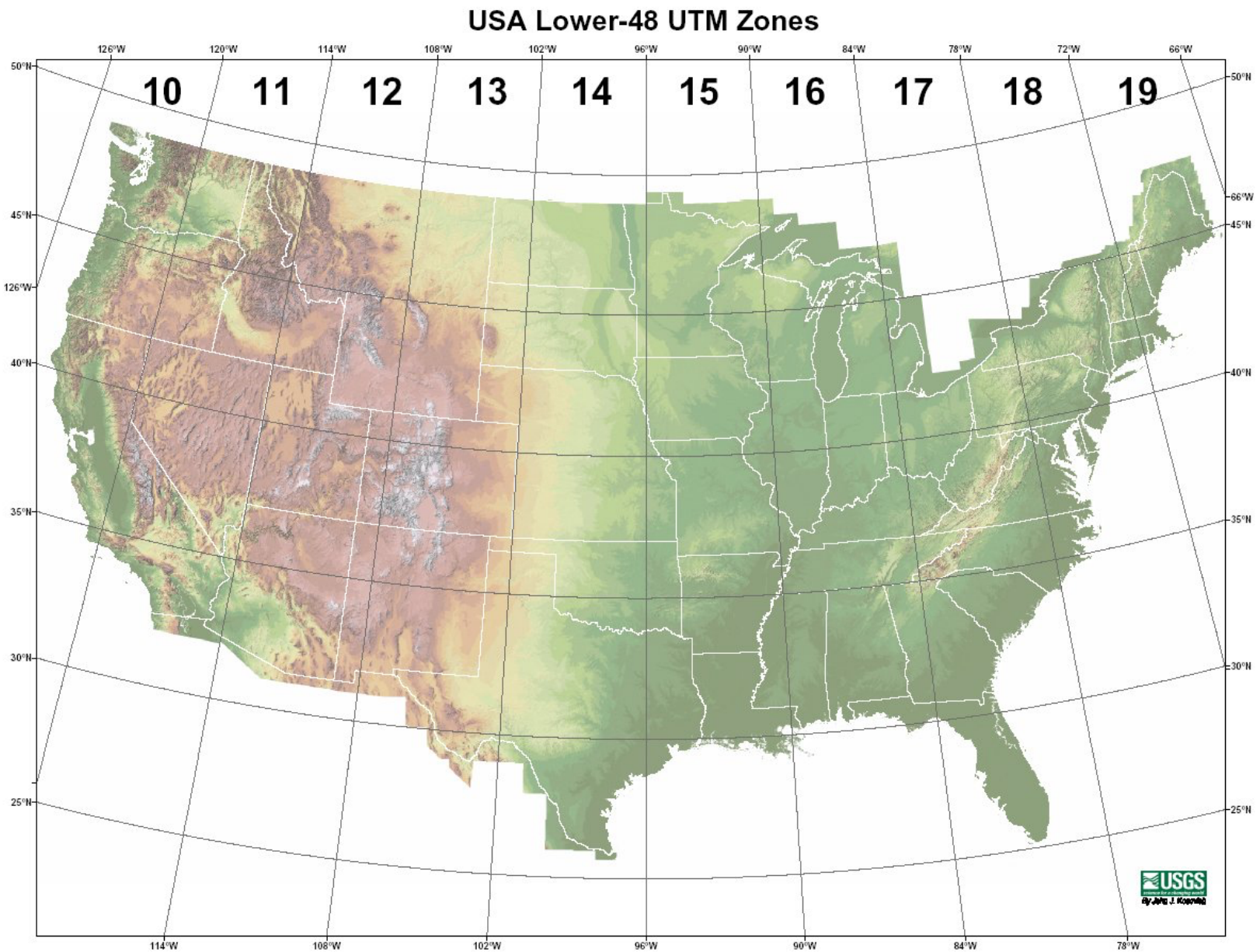
```
1 st_crs(nc)
```

```
Coordinate Reference System:
  User input: NAD83
  wkt:
GEOGCRS["NAD83",
    DATUM["North American Datum 1983",
        ELLIPSOID["GRS 1980",6378137,298.257222101,
            LENGTHUNIT["metre",1]]],
    PRIMEM["Greenwich",0,
        ANGLEUNIT["degree",0.0174532925199433]],
    CS[ellipsoidal,2],
        AXIS["latitude",north,
            ORDER[1],

ANGLEUNIT["degree",0.0174532925199433]],
        AXIS["longitude",east,
            ORDER[2],
```

```
1 st_crs(hwy)
```

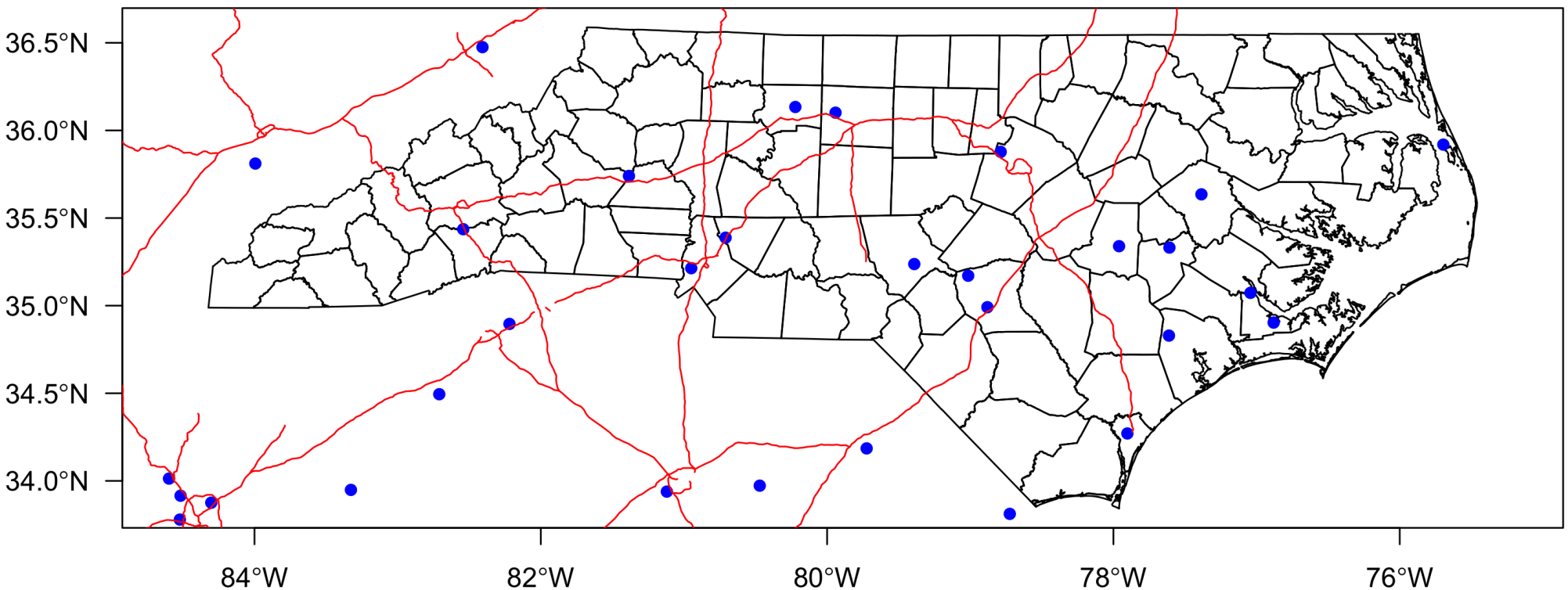
```
Coordinate Reference System:
  User input: NAD83 / UTM zone 15N
  wkt:
PROJCRS["NAD83 / UTM zone 15N",
    BASEGEOGCRS["NAD83",
        DATUM["North American Datum 1983",
            ELLIPSOID["GRS
1980",6378137,298.257222101,
                LENGTHUNIT["metre",1]]],
        PRIMEM["Greenwich",0,

ANGLEUNIT["degree",0.0174532925199433]],
        ID["EPSG",4269]],
    CONVERSION["UTM zone 15N",
        METHOD["Transverse Mercator",
            ID["EPSG",9807]],
        PARAMETER["Latitude of natural origin",0,
```

Aside - UTM Zones



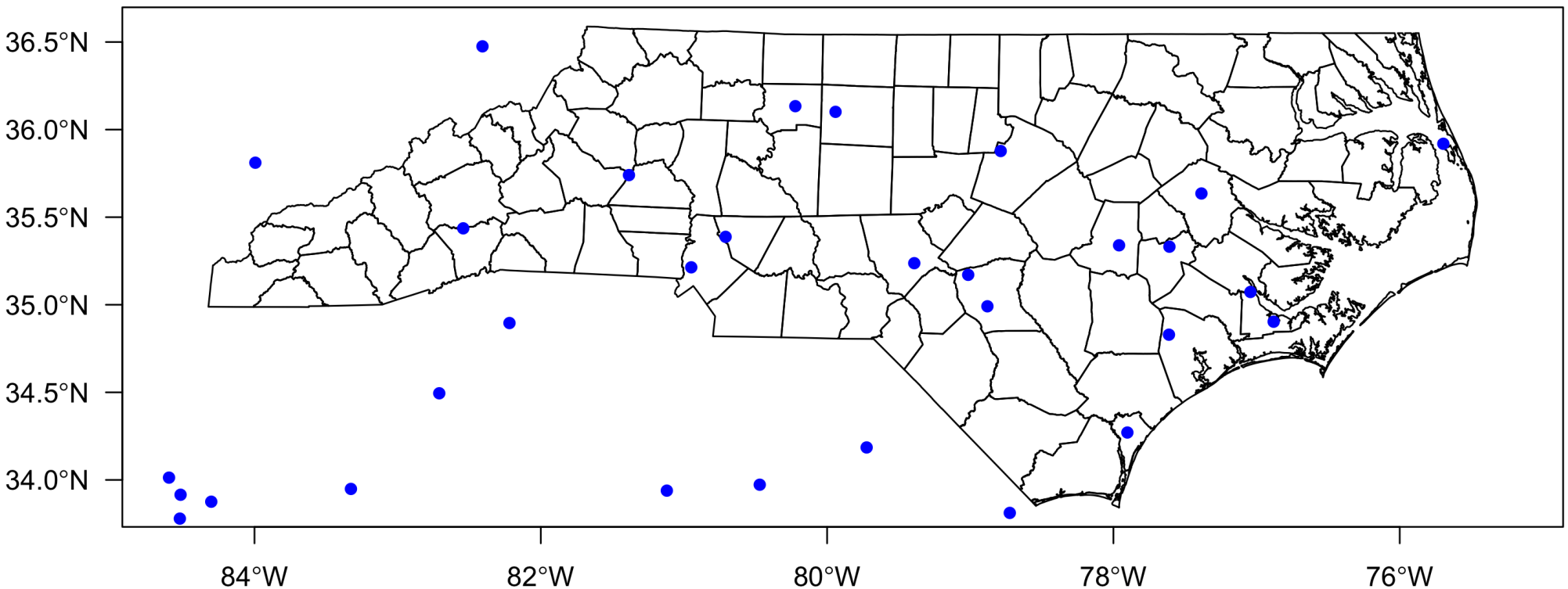
hwy -> Lat/Long

```
1  hwy = st_transform(hwy, st_crs(nc))
```



Airport Example

NC Airports



Sparse Insections

```
1 st_intersects(nc[20:30,], air) %>% str()
```

```
List of 11
 $ : int(0)
 $ : int(0)
 $ : int(0)
 $ : int(0)
 $ : int(0)
 $ : int 268
 $ : int 717
 $ : int(0)
 $ : int(0)
 $ : int(0)
 $ : int(0)
 - attr(*, "predicate")= chr "intersects"
 - attr(*, "region.id")= chr [1:11] "1" "2" "3" "4" ...
 - attr(*, "remove_self")= logi FALSE
 - attr(*, "retain_unique")= logi FALSE
 - attr(*, "ncol")= int 940
```

Dense Insections

```
1 st_intersects(nc, air, sparse=FALSE) %>% str()
```

```
logi [1:100, 1:940] FALSE FALSE FALSE FALSE FALSE FALSE ...
```

```
1 st_intersects(nc, air, sparse=FALSE) %>% .[20:30, 260:270]
```

```
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]
 [1,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [2,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [3,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [4,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [5,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [6,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [7,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [8,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [9,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[10,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[11,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
      [,8]  [,9] [,10] [,11]
 [1,] FALSE FALSE FALSE FALSE
 [2,] FALSE FALSE FALSE FALSE
 [3,] FALSE FALSE FALSE FALSE
 [4,] FALSE FALSE FALSE FALSE
```

Which counties have airports?

```
nc_air = nc %>%
  mutate(
    n_air = map_int(
      st_intersects(nc, air),
      length
    )
  ) %>%
  filter(n_air > 0)

nc_air %>% pull(COUNTY)
```

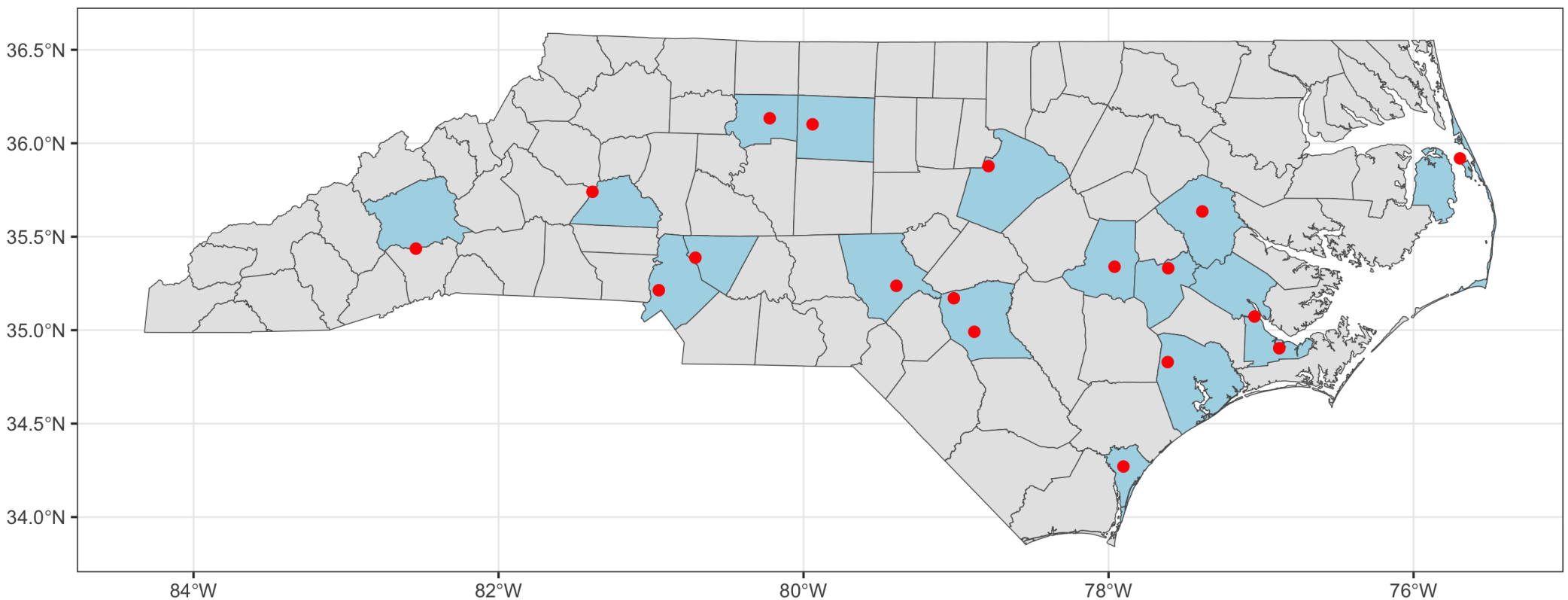
```
 [1] "Forsyth County"     "Guilford County"
 [3] "Dare County"        "Wake County"
 [5] "Pitt County"        "Catawba County"
 [7] "Buncombe County"    "Wayne County"
 [9] "Mecklenburg County" "Moore County"
[11] "Cabarrus County"    "Lenoir County"
[13] "Craven County"      "Cumberland County"
[15] "Onslow County"      "New Hanover County"
```

```
air_nc = air %>%
  slice(
    st_intersects(nc, air) %>%
      unlist() %>%
      unique()
  )
air_nc %>% pull(AIRPT_NAME)
```

```
 [1] "SMITH REYNOLDS AIRPORT"
 [2] "PIEDMONT TRIAD INTERNATIONAL AIRPORT"
 [3] "DARE COUNTY REGIONAL AIRPORT"
 [4] "RALEIGH-DURHAM INTERNATIONAL AIRPORT"
 [5] "PITT-GREENVILLE AIRPORT"
 [6] "HICKORY REGIONAL AIRPORT"
 [7] "ASHEVILLE REGIONAL AIRPORT"
 [8] "SEYMOUR JOHNSON AIR FORCE BASE"
 [9] "CHARLOTTE/DOUGLAS INTERNATIONAL AIRPORT"
[10] "MOORE COUNTY AIRPORT"
[11] "CONCORD REGIONAL AIRPORT"
[12] "KINSTON REGIONAL JETPORT AT STALLINGS FIELD"
[13] "CHERRY POINT MARINE CORPS AIR STATION
/CUNNINGHAM FIELD/"
[14] "COASTAL CAROLINA REGIONAL AIRPORT"
[15] "POPE AIR FORCE BASE"
[16] "FAYETTEVILLE REGIONAL/GRANNIS FIELD"
```

Results

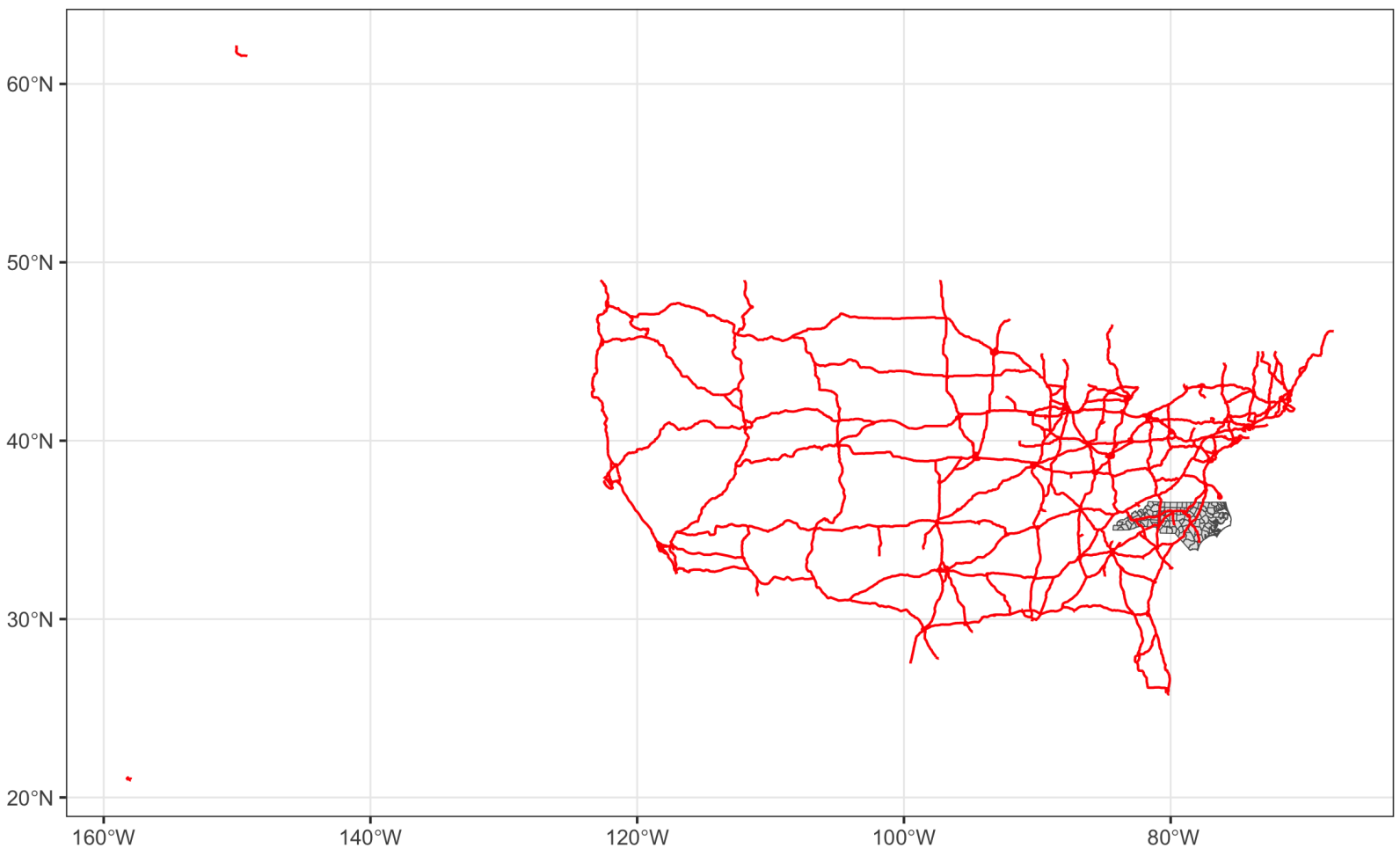
```
ggplot() +
  geom_sf(data=nc) +
  geom_sf(data = nc_air, fill = "lightblue") +
  geom_sf(data = air_nc, color = "red", size=2)
```



Highway Example

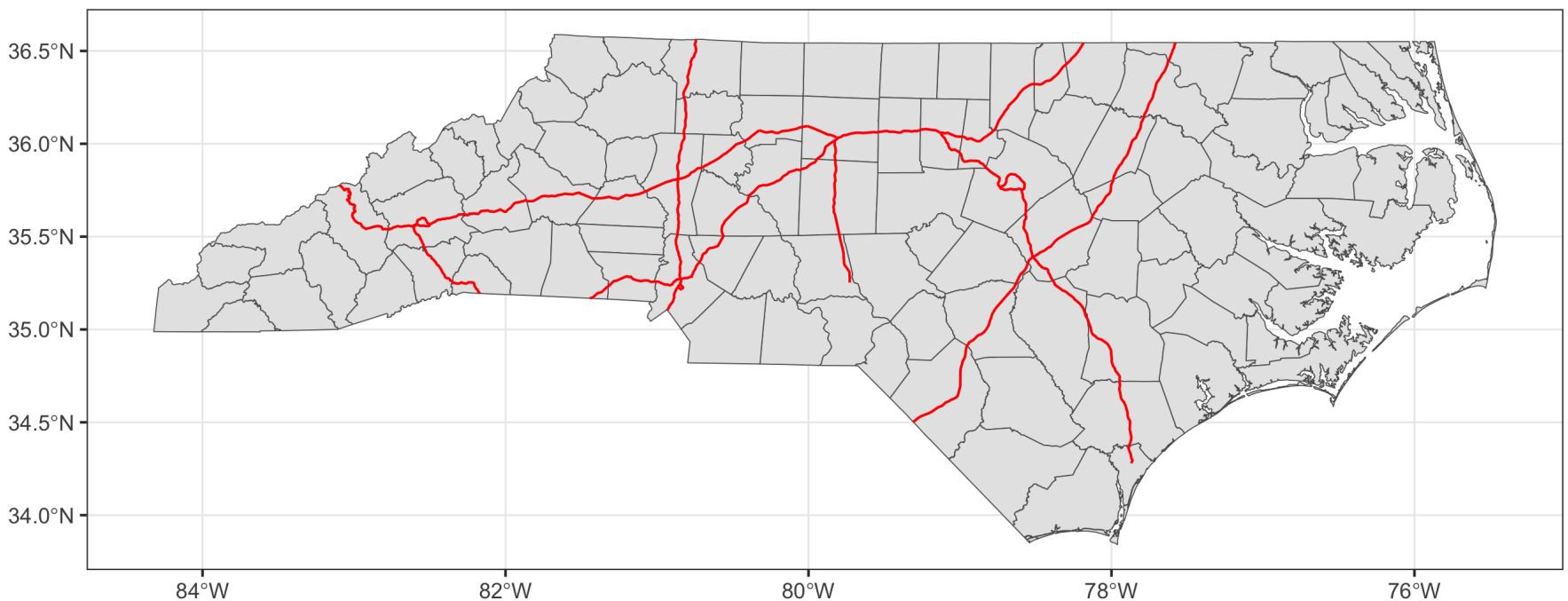
Highways

```
ggplot() +
  geom_sf(data=nc) +
  geom_sf(data=hwy, col='red')
```



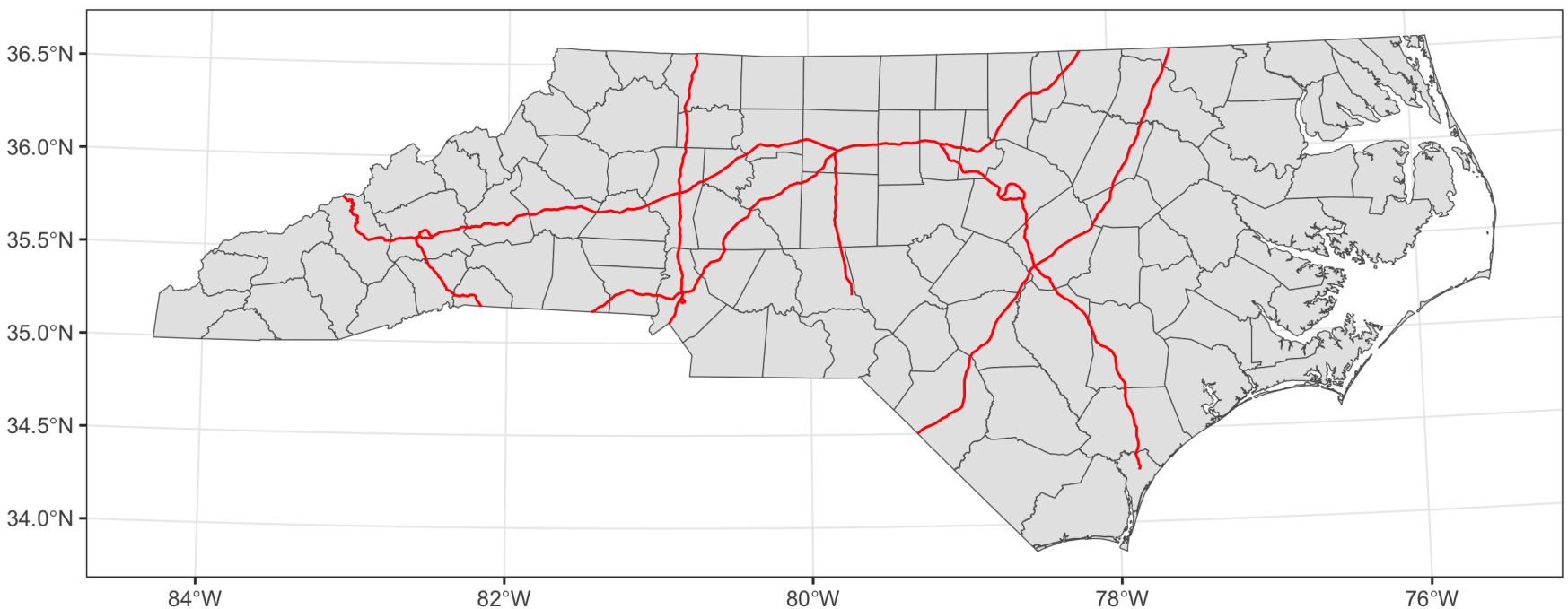
NC Interstate Highways

```
hwy_nc = st_intersection(hwy, nc)

ggplot() +
  geom_sf(data=nc) +
  geom_sf(data=hwy_nc, col='red')
```



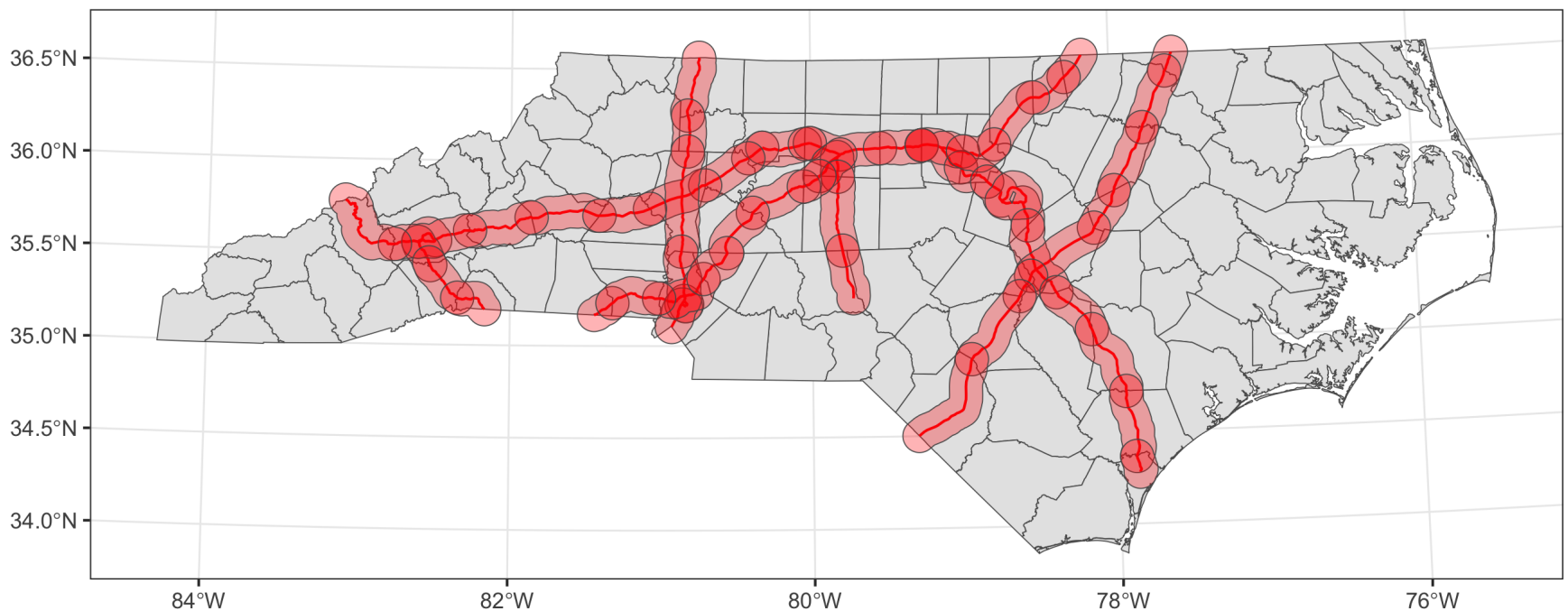
Counties near a highway (Projection)

```
nc_utm  = st_transform(nc, "+proj=utm +zone=17 +datum=NAD83 +units=m +no_defs")
hwy_utm  = st_transform(hwy, "+proj=utm +zone=17 +datum=NAD83 +units=m +no_defs")

hwy_nc = st_intersection(hwy_utm, nc_utm)

ggplot() +
  geom_sf(data=nc_utm) +
  geom_sf(data=hwy_nc, col='red')
```



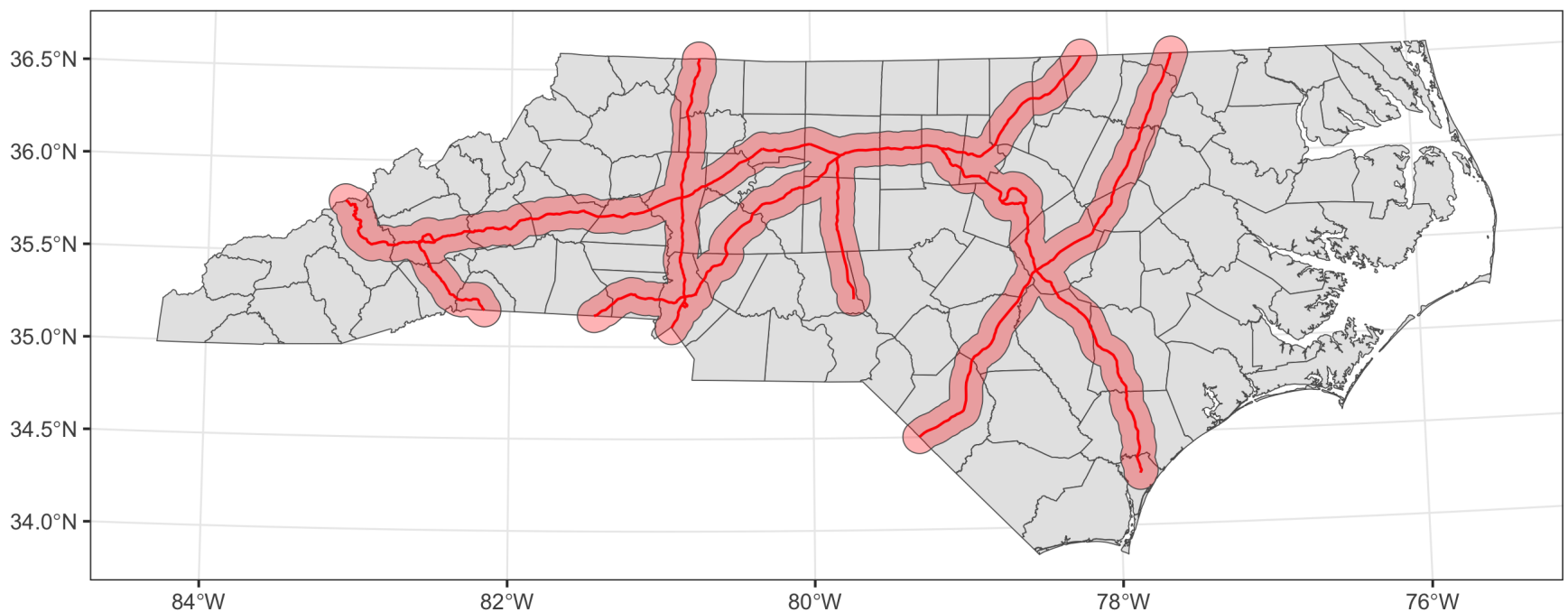
Counties near the interstate (Buffering)

```
hwy_nc_buffer = hwy_nc %>%
  st_buffer(10000)

ggplot() +
  geom_sf(data=nc_utm) +
  geom_sf(data=hwy_nc, color='red') +
  geom_sf(data=hwy_nc_buffer, fill='red', alpha=0.3)
```

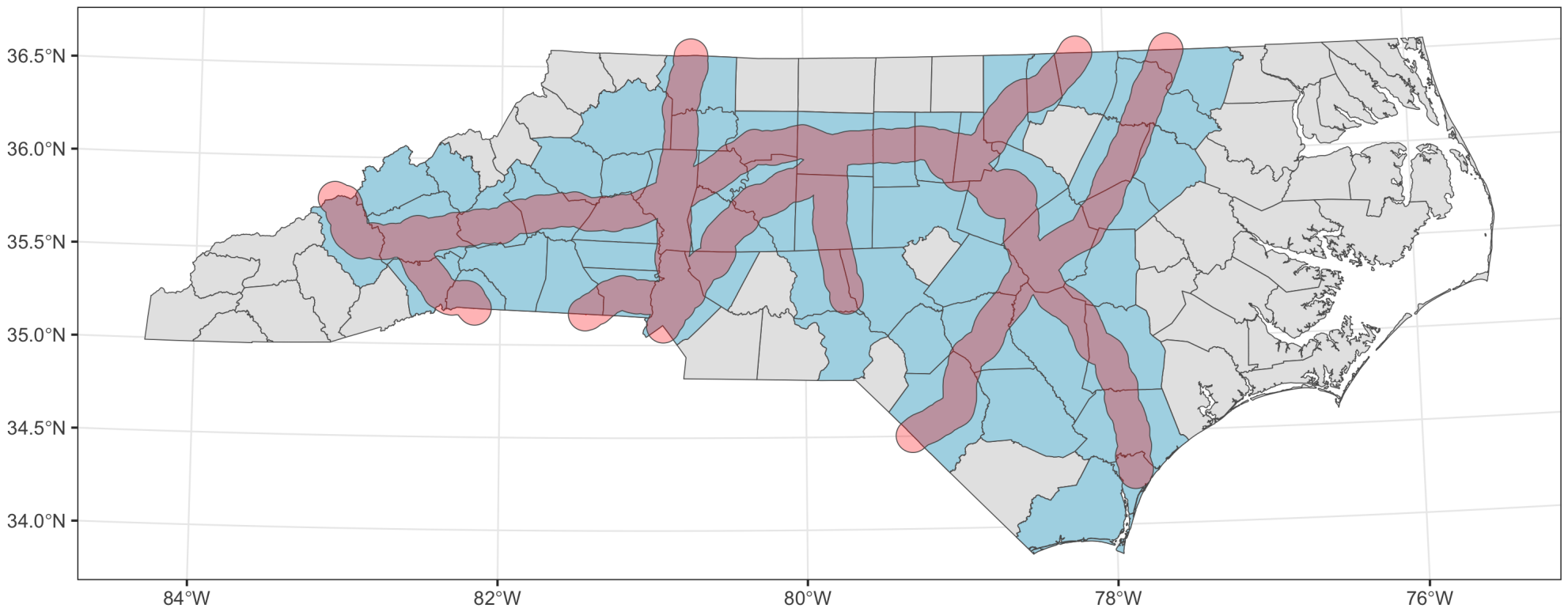


Counties near the interstate (Buffering + Union)

```
hwy_nc_buffer = hwy_nc %>%
  st_buffer(10000) %>%
  st_union() %>%
  st_sf()

ggplot() +
  geom_sf(data=nc_utm) +
  geom_sf(data=hwy_nc, color='red') +
  geom_sf(data=hwy_nc_buffer, fill='red', alpha=0.3)
```



Counties near the interstate (Buffering + Union)



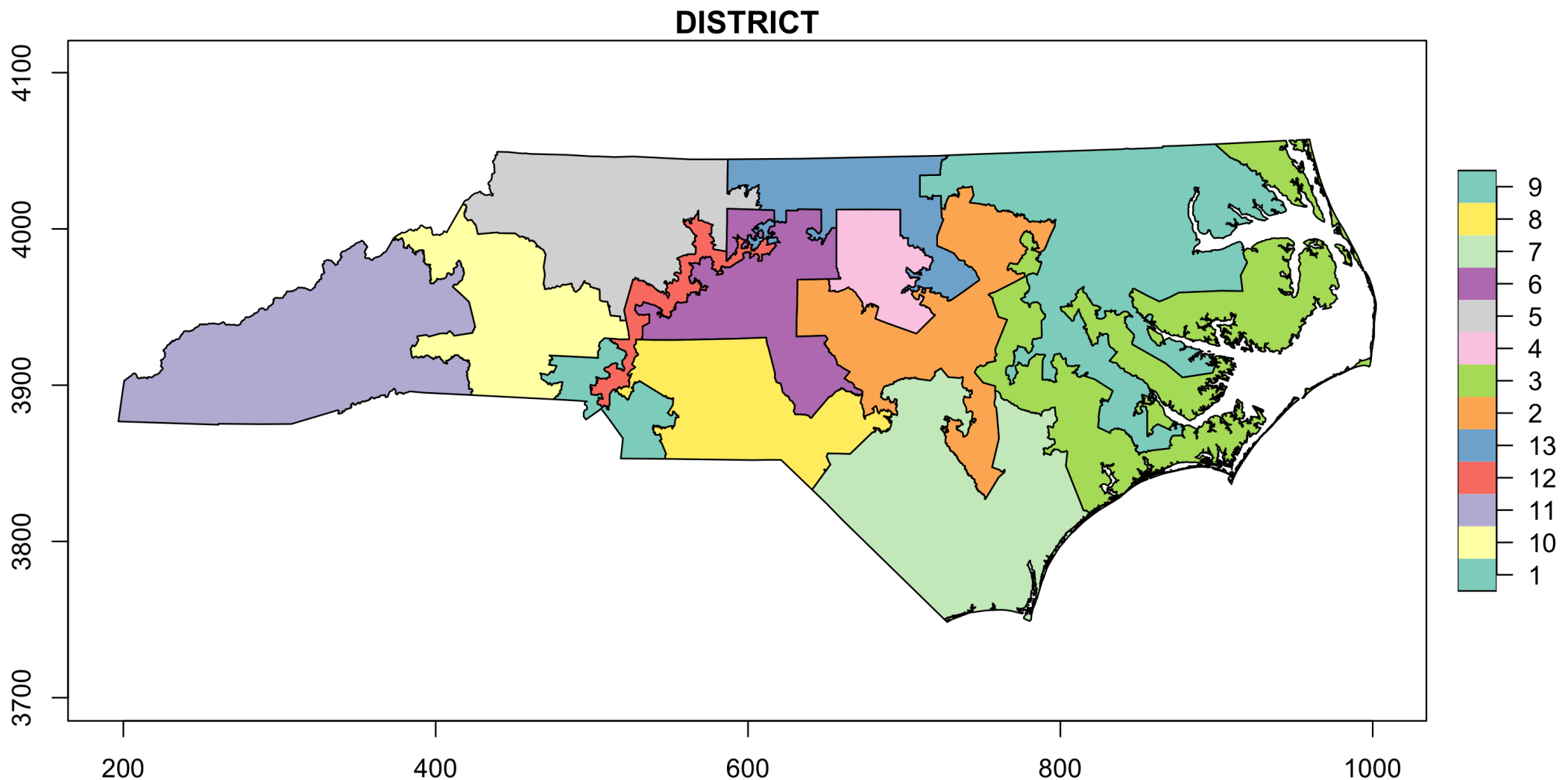
Gerrymandering Example

NC House Districts - 112th Congress

```
( nc_house = read_sf("data/gis/nc_districts112.gpkg", quiet = TRUE) |>
    select(ID, DISTRICT) )
```

```
Simple feature collection with 13 features and 2 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: -84.32187 ymin: 33.84452 xmax: -75.45998 ymax: 36.58812
Geodetic CRS:  WGS 84
# A tibble: 13 × 3
   ID           DISTRICT                       geom
   <chr>        <chr>                <MULTIPOLYGON [°]>
 1 037108112001 1        (((-77.32845 35.35031, -…
 2 037108112002 2        (((-78.89928 35.12619, -…
 3 037108112003 3        (((-75.68266 35.23291, -…
 4 037108112004 4        (((-78.77926 35.78568, -…
 5 037108112005 5        (((-79.8968 36.38075, -7…
 6 037108112006 6        (((-80.4201 35.68953, -8…
```

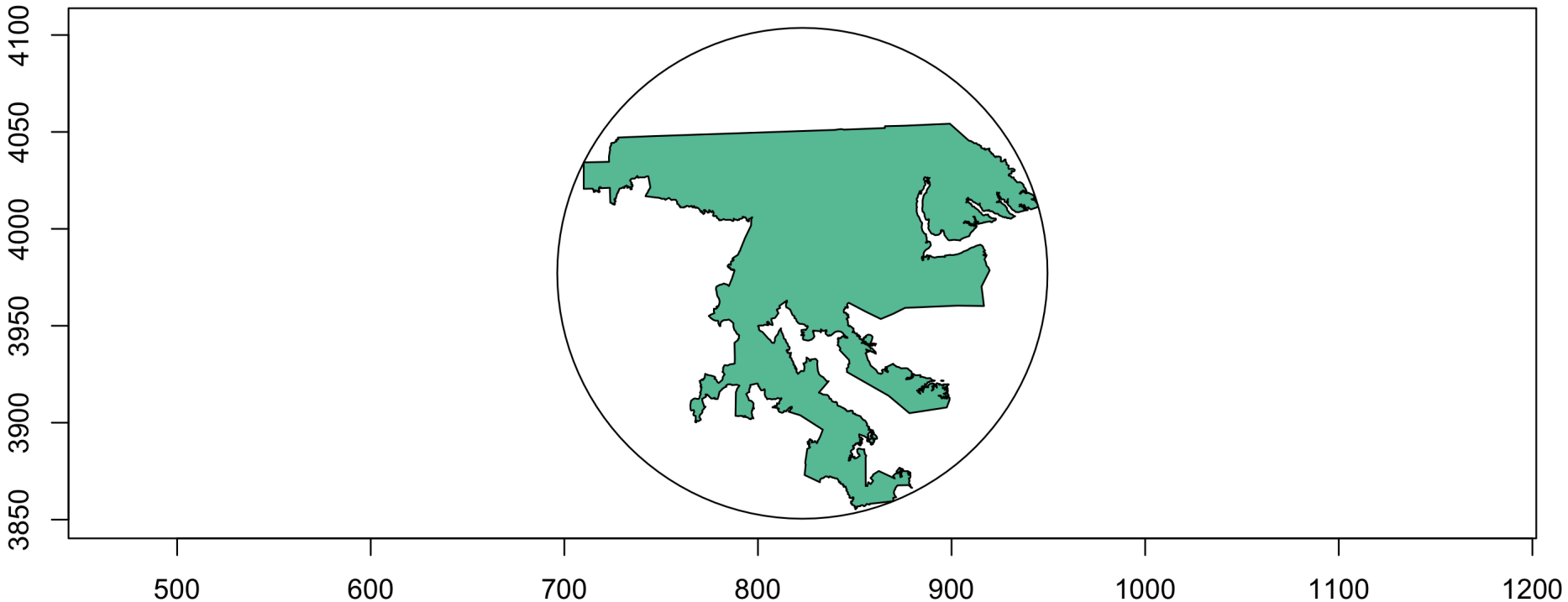
```r
nc_house = nc_house |>
  st_transform("+proj=utm +zone=17 +datum=NAD83 +units=km +no_defs")
plot(nc_house[,"DISTRICT"], axes=TRUE)
```



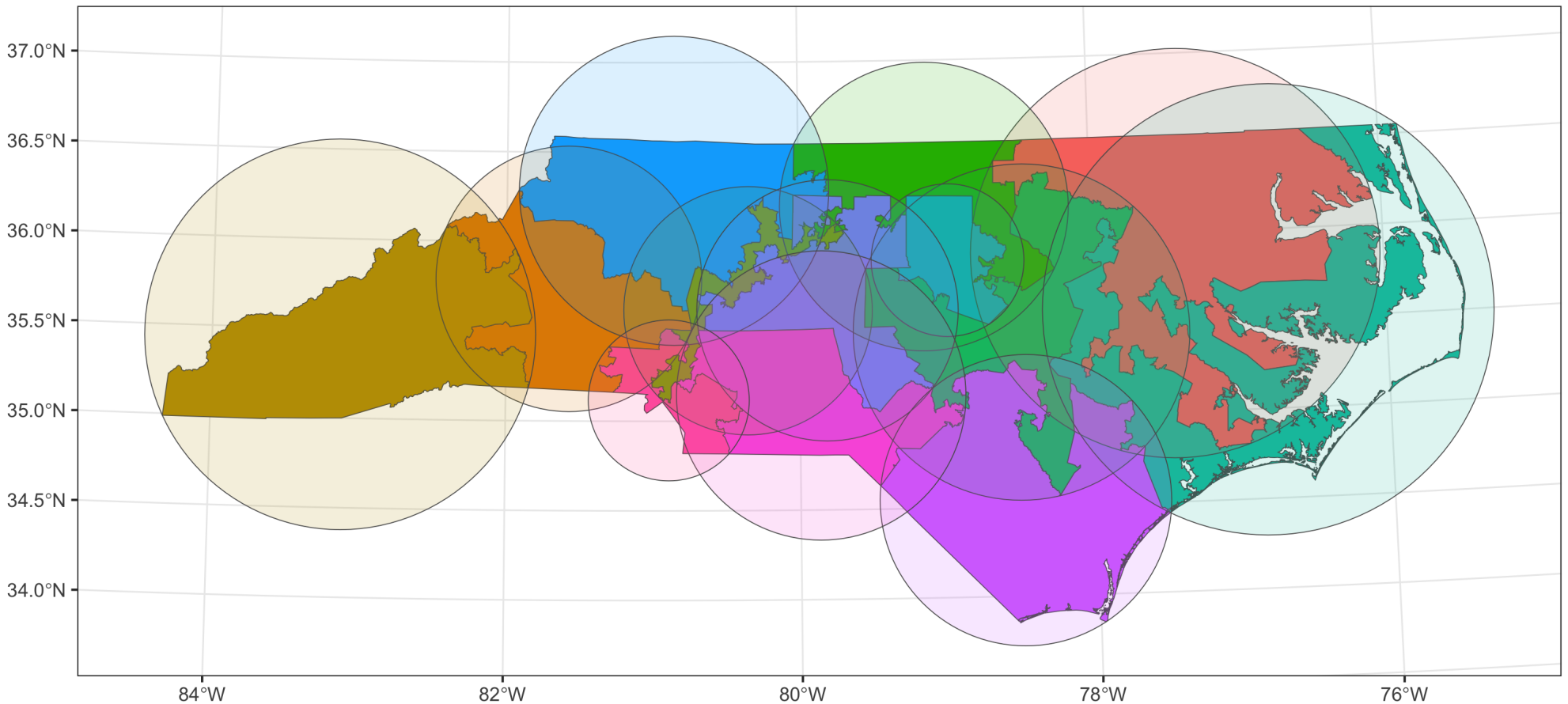
Measuring Compactness - Reock Score

The Reock score is a measure of compactness that is calculated as the ratio of a shape's area to the area of its minimum bounding circle.

```
circs = nc_house |>
  lwgeom::st_minimum_bounding_circle()
plot(circs |> filter(DISTRICT == 1) |> st_geometry(), axes=TRUE)
plot(nc_house |> select(DISTRICT) |> filter(DISTRICT == 1), add=TRUE)
```

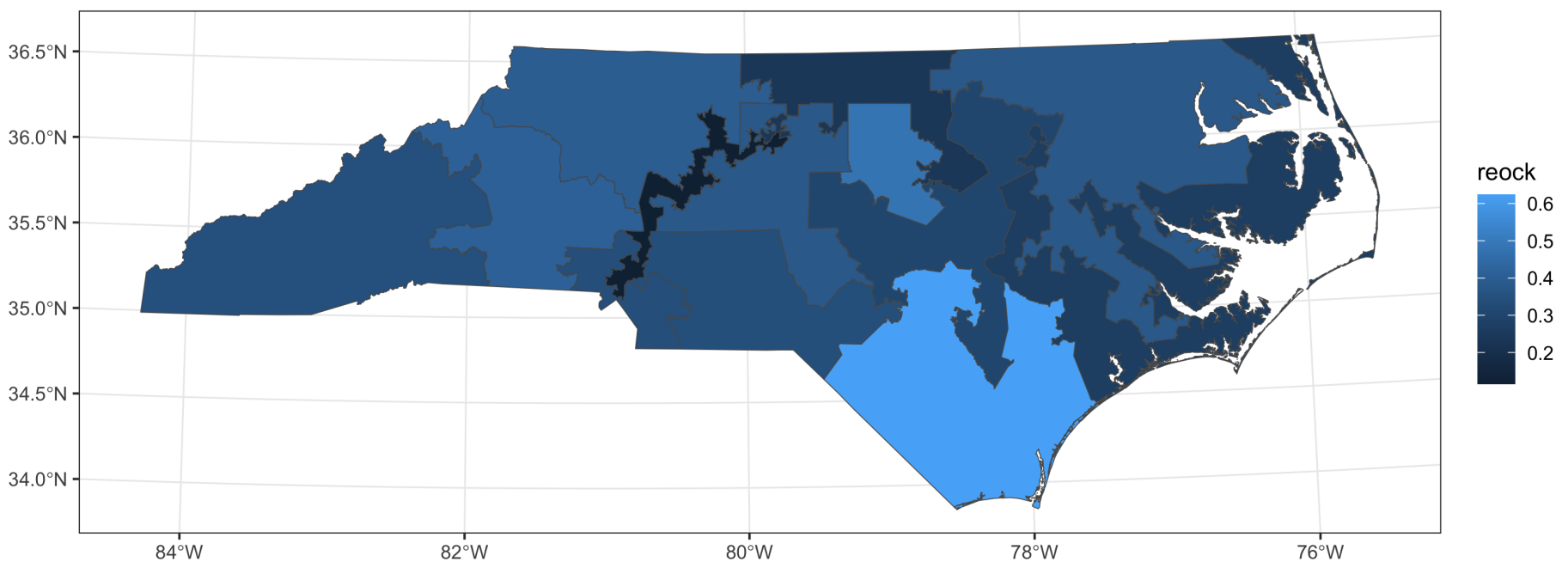

4100
4050
4000
3950
3900
3850
500
600
700
800
900
1000
1100
1200

```
ggplot(mapping = aes(fill=DISTRICT)) +
  geom_sf(data=nc_house) +
  geom_sf(data=circs, alpha=0.15) +
  guides(color="none", fill="none")
```



Calculating Reock

```
nc_house |>
  mutate(reock = (st_area(nc_house) / st_area(circs)) |> as.numeric()) |>
  ggplot(aes(fill = reock)) +
    geom_sf()
```



```r
nc_house |>
  mutate(reock = st_area(nc_house) / st_area(circs)) |>
  arrange(reock) |>
  print(n=Inf)
```

```
Simple feature collection with 13 features and 3 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: 196.7724 ymin: 3748.368 xmax: 1002.17 ymax: 4057.317
CRS:           +proj=utm +zone=17 +datum=NAD83 +units=km +no_defs
# A tibble: 13 × 4
   ID      DISTR…¹                      geom reock
   <chr>   <chr>         <MULTIPOLYGON [km]>   [1]
 1 037108… 12      (((545.2987 3945.168, 54… 0.116
 2 037108… 13      (((597.9665 4026.852, 59… 0.237
 3 037108… 3       (((984.0709 3911.853, 98… 0.266
 4 037108… 2       (((691.4141 3889.056, 69… 0.303
 5 037108… 9       (((506.3207 3893.208, 50… 0.339
 6 037108… 8       (((688.6584 3870.456, 68… 0.342
 7 037108… 11      (((226.7522 3918.52, 226… 0.344
 8 037108… 6       (((552.4691 3949.669, 55… 0.378
 9 037108… 1       (((833.677 3918.083, 831… 0.378
```