

# Spatial Data and Cartography (Part 2)

Lecture 17

Dr. Colin Rundel

# `sf` object summary

- `sf` class is an extension of `data.frame` / `tibble` that includes a `geometry` column
- The `geometry` column is a list column with the `sfc` class
  - This column also tracks the CRS of the geometry (set via `st_crs()` or transformed by `st_transform()`)
- Individual elements of the `geometry` column are objects with the `sfg` class
  - S3 class also contains the simple feature geometry type and coordinate type

# Plotting

# Example Data - NC SIDS

```
1 ( nc = read_sf(system.file("shape/nc.shp", package="sf"), quiet = TRUE) |>
2   select(-(AREA:CNTY_ID), -(FIPS:CRESS_ID)))
```

Simple feature collection with 100 features and 7 fields

Geometry type: MULTIPOLYGON

Dimension: XY

Bounding box: xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax: 36.58965

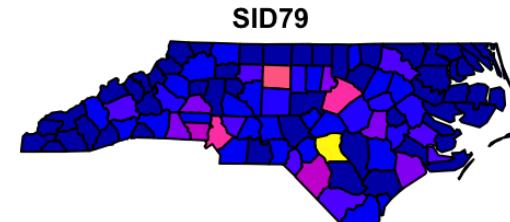
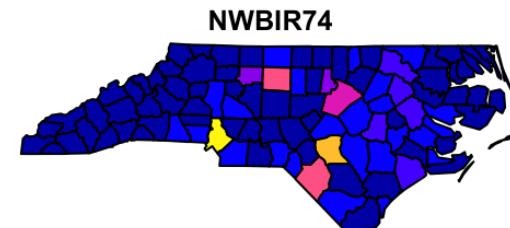
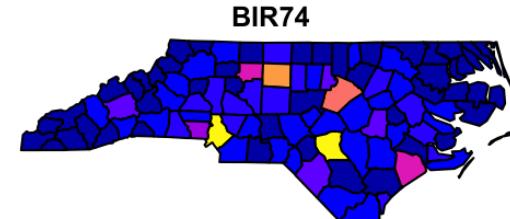
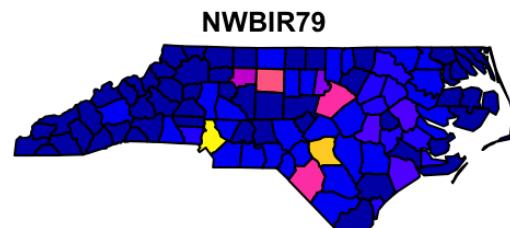
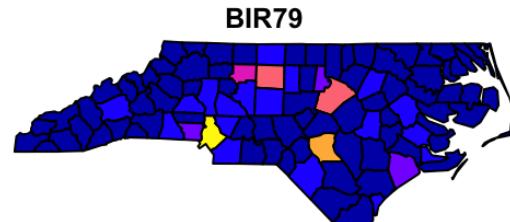
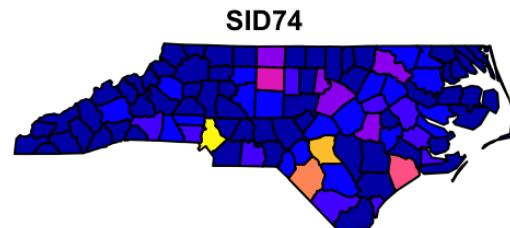
Geodetic CRS: NAD27

# A tibble: 100 × 8

	NAME	BIR74	SID74	NWBIR74	BIR79	SID79	NWBIR79
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	Ashe	1091	1	10	1364	0	19
2	Alleghany	487	0	10	542	3	12
3	Surry	3188	5	208	3616	6	260
4	Currituck	508	1	123	830	2	145
5	Northhampton	1421	9	1066	1606	3	1197
6	Hertford	1452	7	954	1838	5	1237
7	Camden	286	0	115	350	2	139
8	Gates	420	0	254	594	2	371
9	Warren	968	4	748	1190	2	844
10	Stokes	1612	1	160	2038	5	176
# i 90 more rows							
# i 1 more variable: geometry <MULTIPOLYGON [°]>							

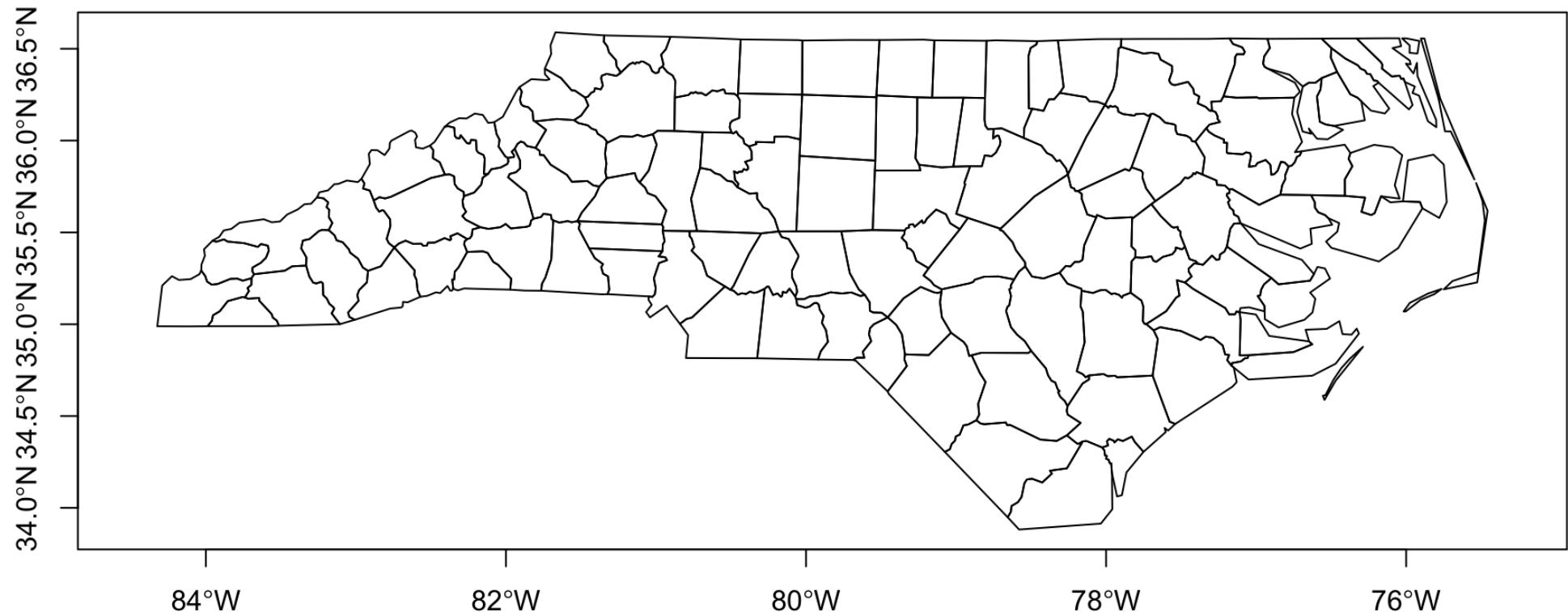
# Base Plots

```
1 plot(nc)
```



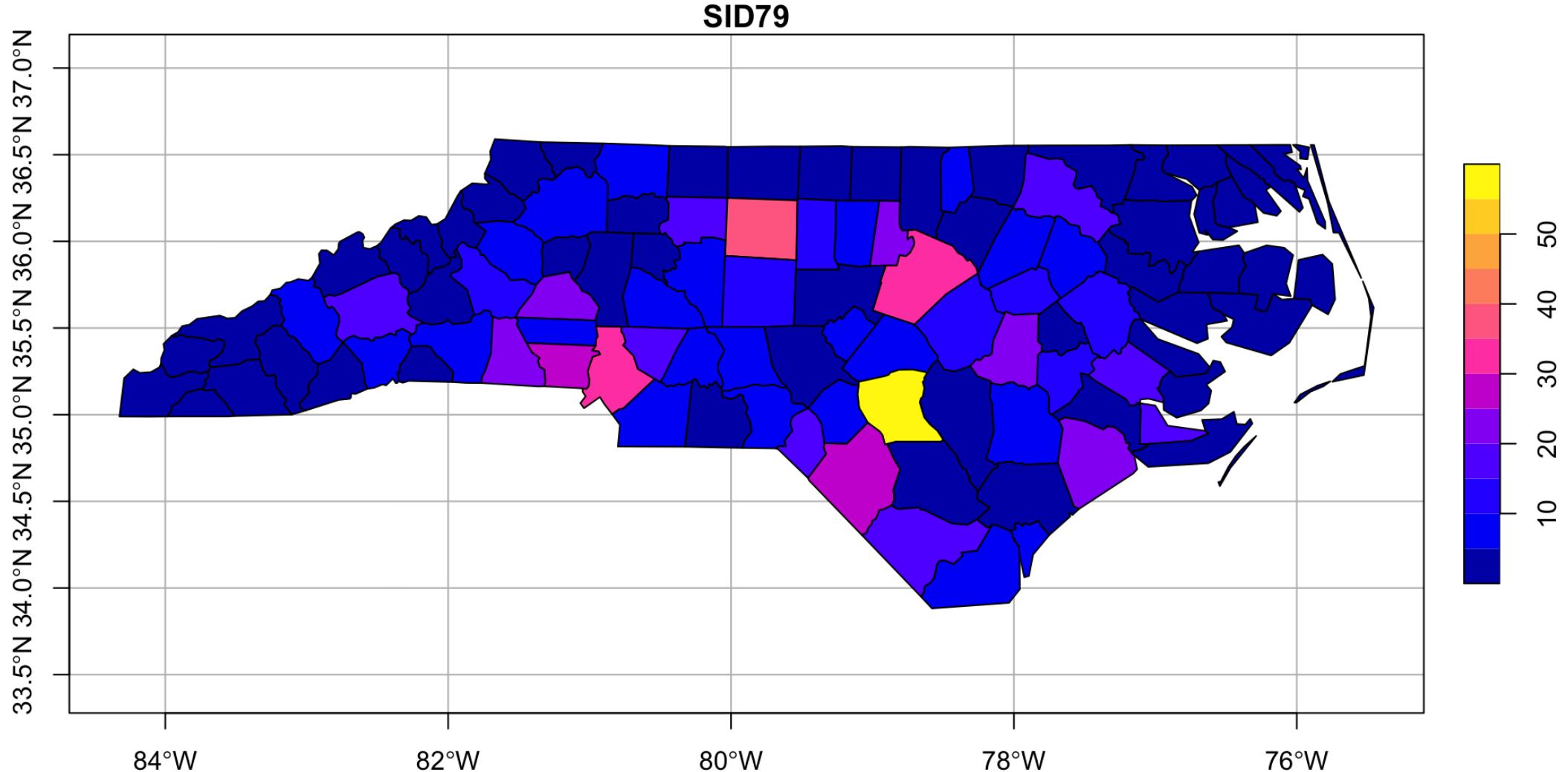
# Geometry Plot

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



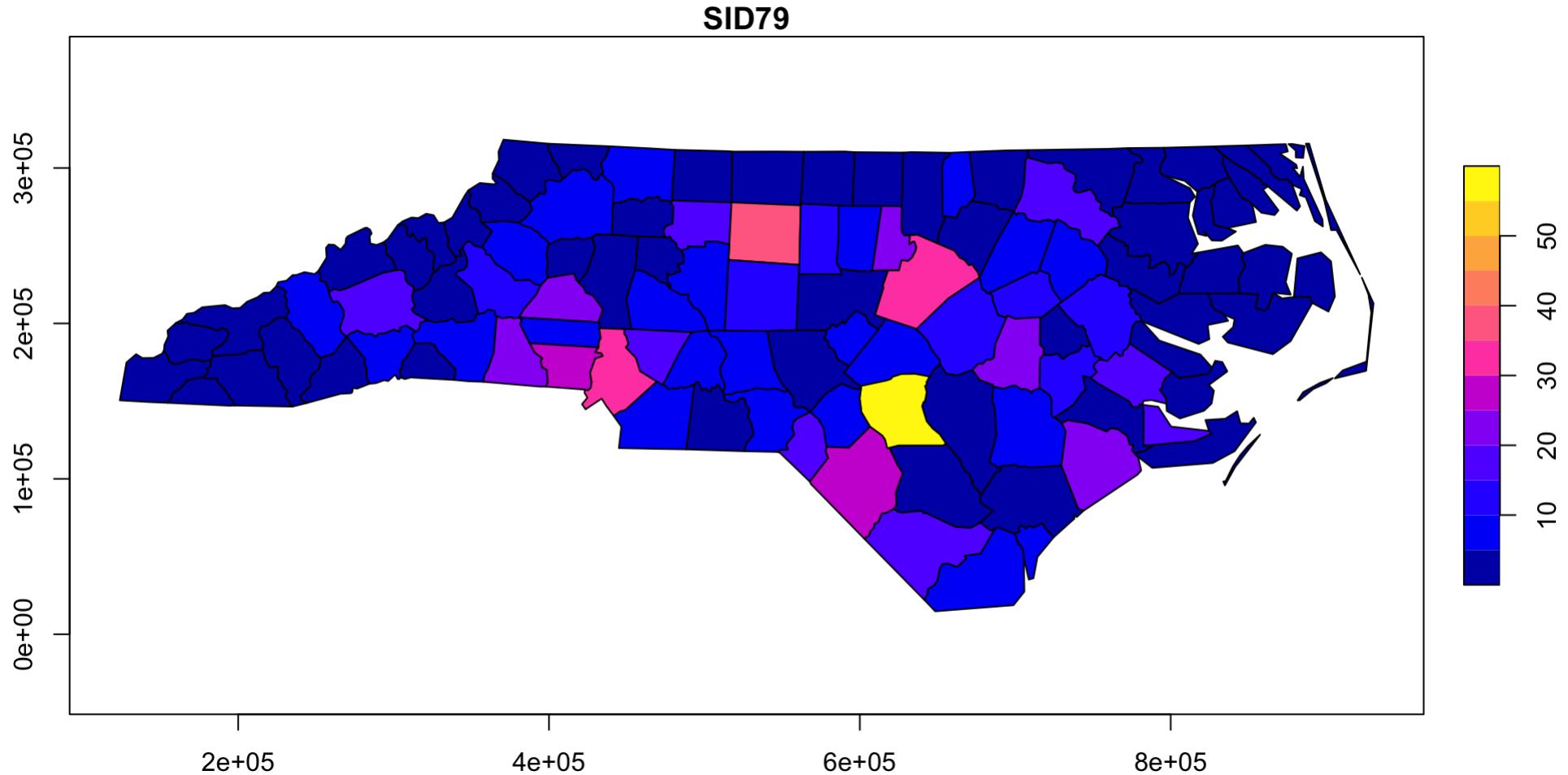
# Graticules

```
1 plot(nc[, "SID79"], graticule=TRUE, axes=TRUE)
```



# EPSG 3631

```
1 plot(st_transform(nc[, "SID79"], 3631), axes=TRUE)
```

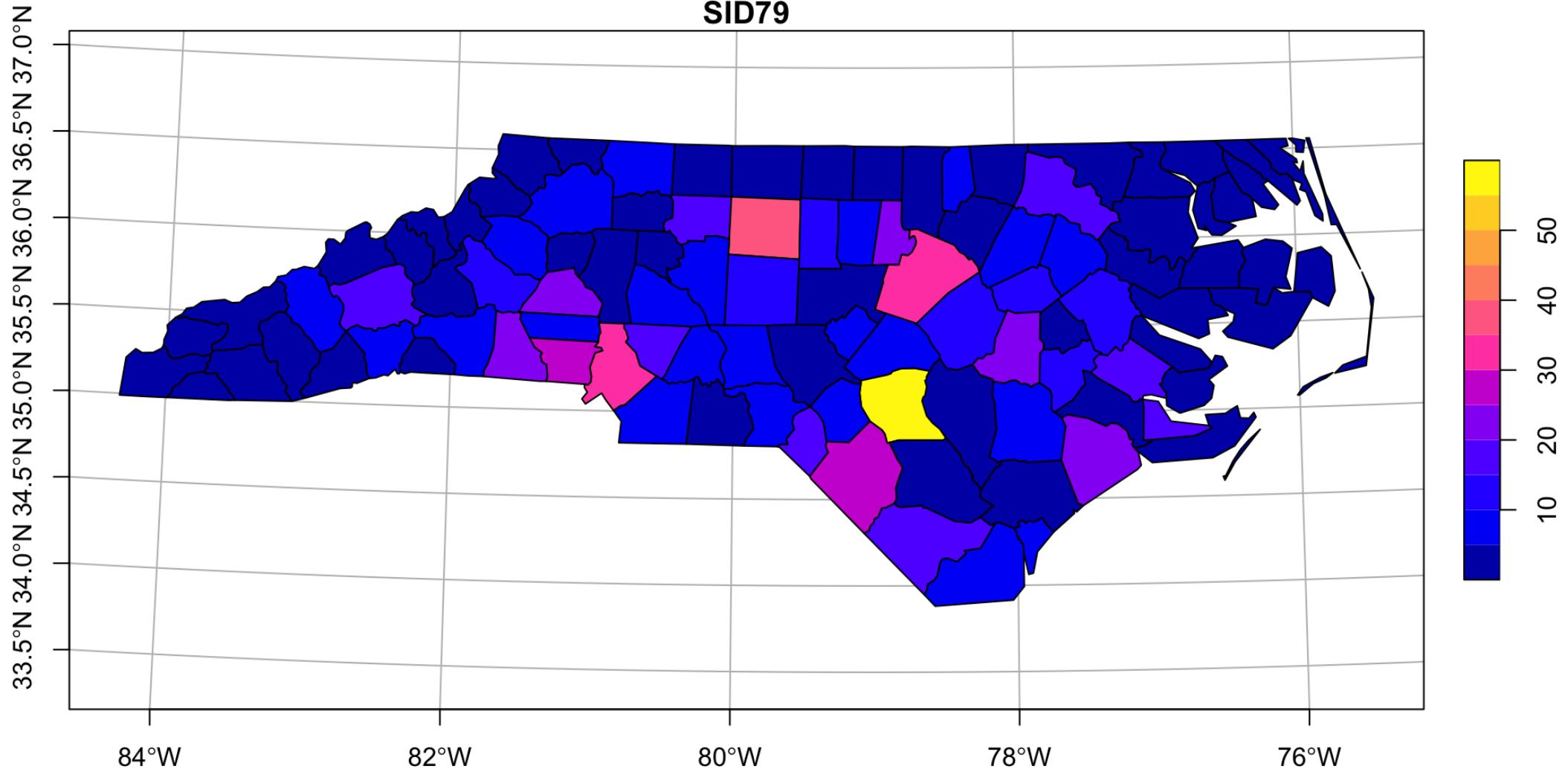


EPSG 3631 is a projected LCC coordinate system that is centered on North Carolina (uses meters as units).

Sta 344/644 - Fall 2023

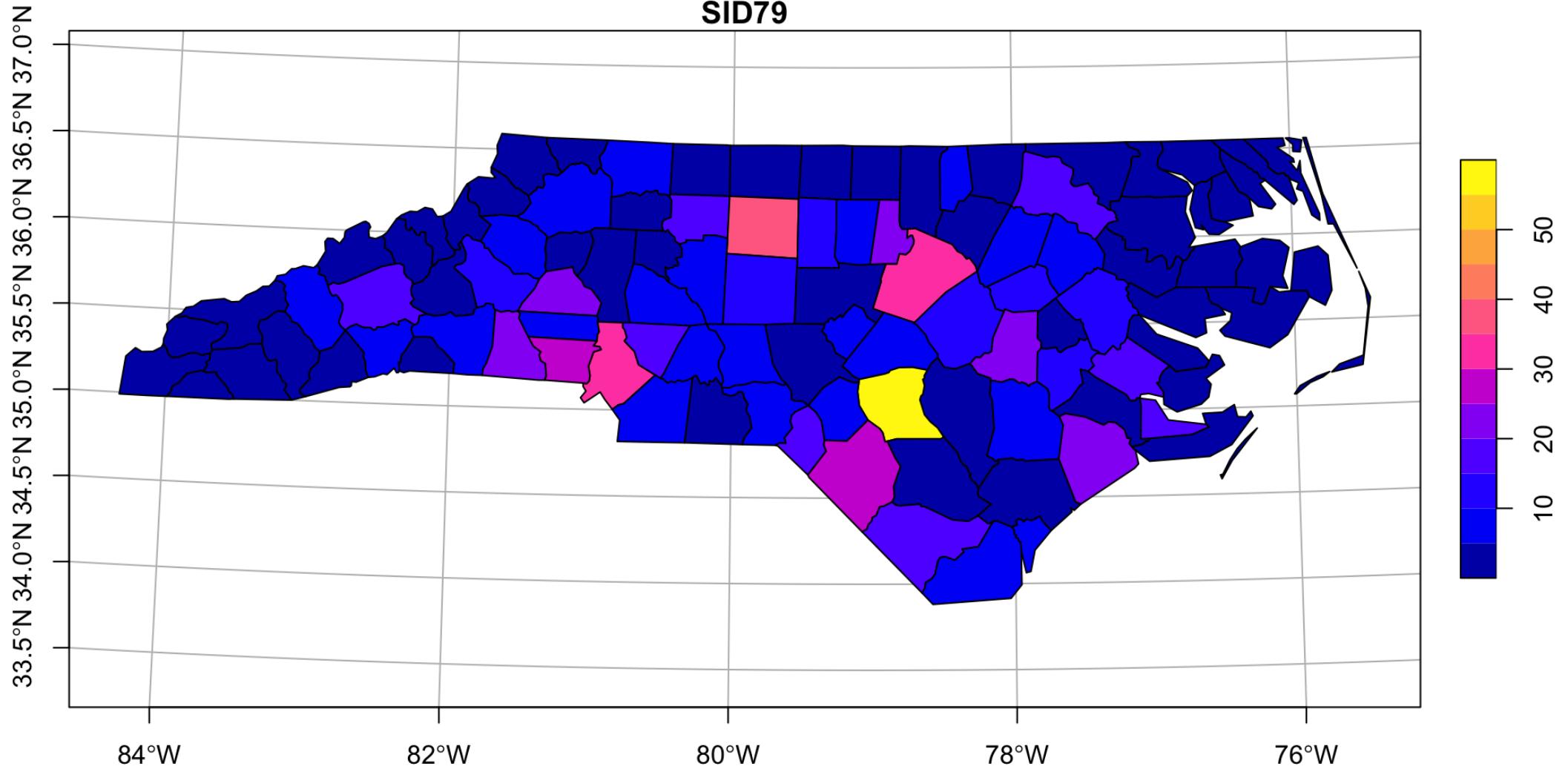
# EPSG 3631 w/ Graticules

```
1 plot(st_transform(nc[, "SID79"], 3631), graticule=TRUE, axes=TRUE)
```



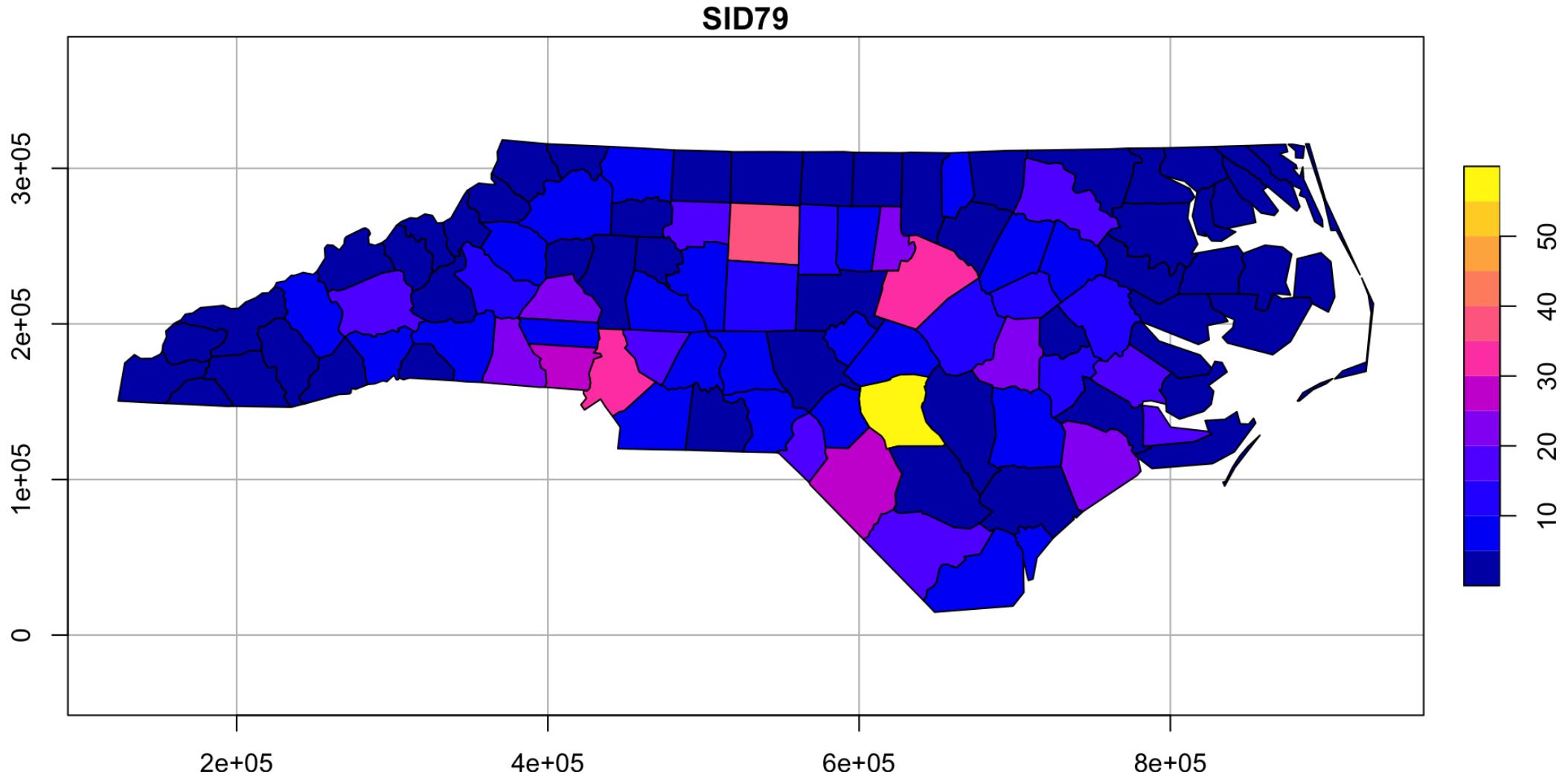
# EPSG 3631 w/ Lat / long Graticules

```
1 plot(st_transform(nc[, "SID79"], 3631), graticule=st_crs(4326), axes=TRUE)
```



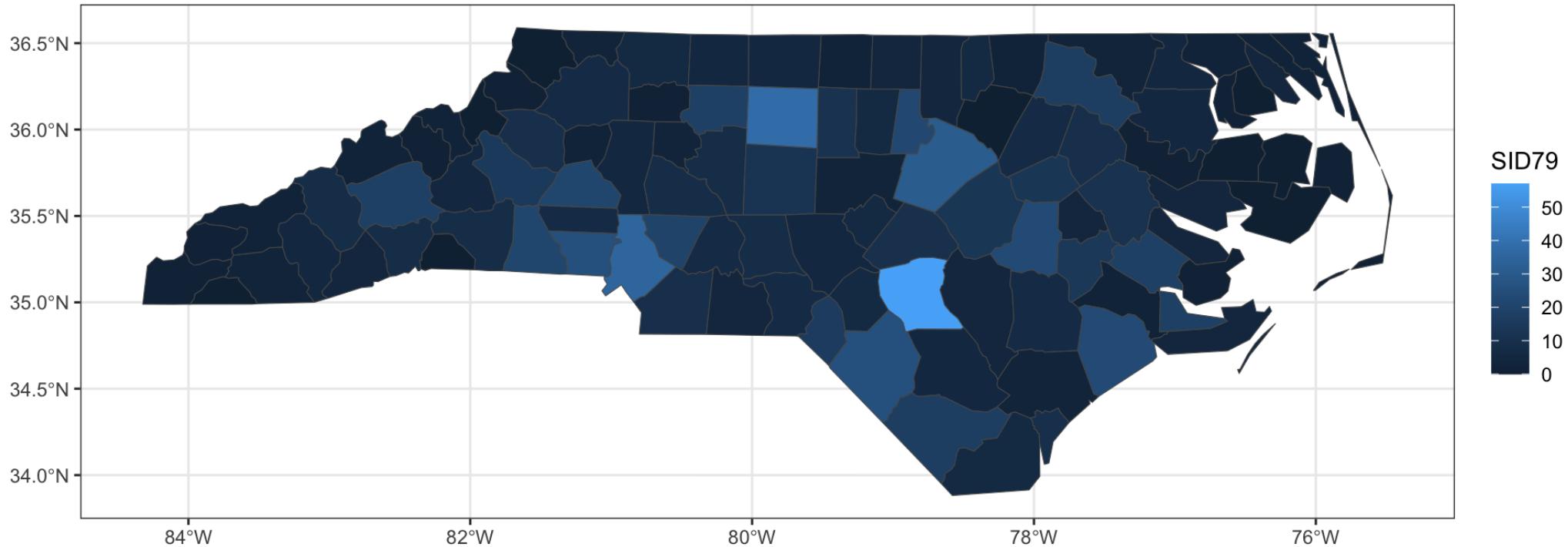
# EPSG 3631 w/ 3631 Graticules

```
1 plot(st_transform(nc[, "SID79"], 3631), graticule=st_crs(3631), axes=TRUE)
```



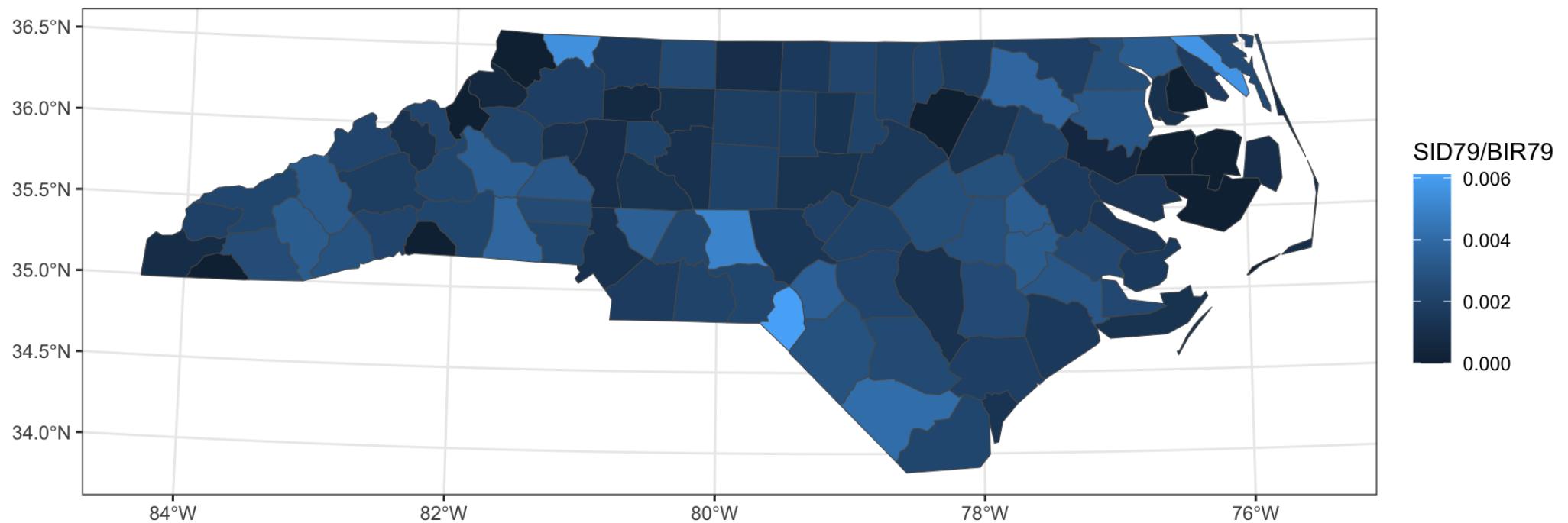
# ggplot2

```
1 ggplot(nc) +  
2   geom_sf(aes(fill=SID79))
```



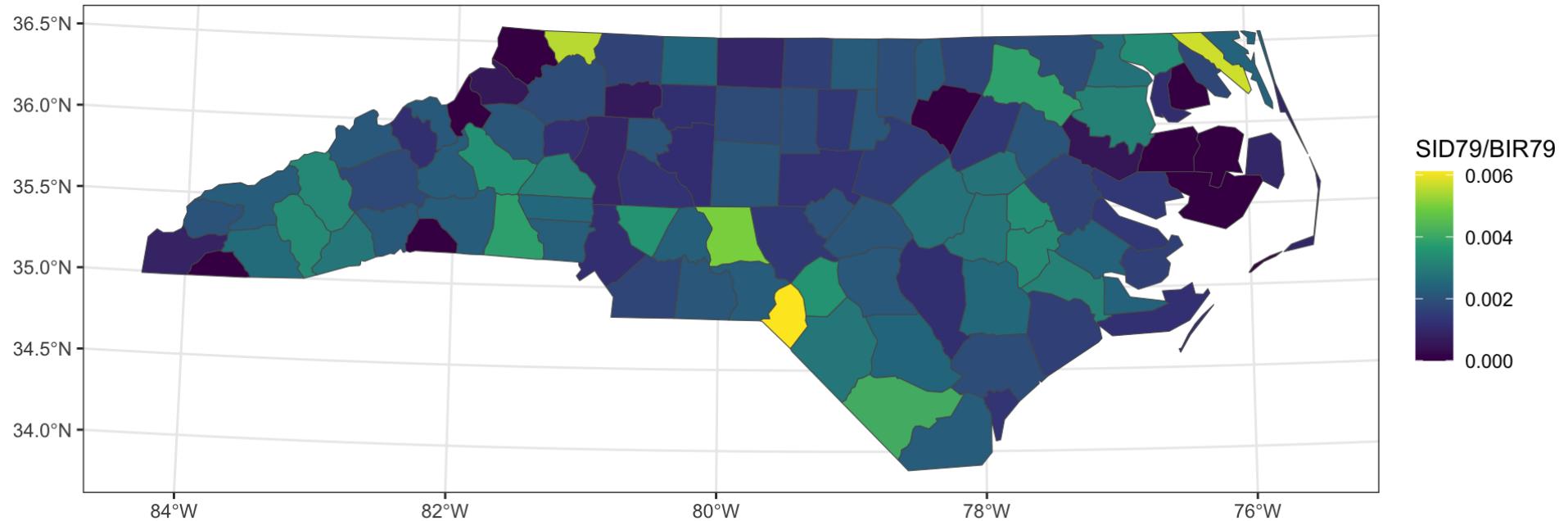
# ggplot2 + projections

```
1 ggplot(st_transform(nc, 3631)) +  
2   geom_sf(aes(fill=SID79 / BIR79))
```



# ggplot2 + viridis

```
1 ggplot(st_transform(nc, 3631)) +  
2   geom_sf(aes(fill=SID79 / BIR79)) +  
3   scale_fill_viridis_c()
```



# Example Data - Meuse

```
1 data(meuse, meuse.riv, package="sp")
2 (meuse = st_as_sf(meuse, coords=c("x", "y"), crs=28992) |>
3   as_tibble() |> st_as_sf())
```

Simple feature collection with 155 features and 12 fields

Geometry type: POINT

Dimension: XY

Bounding box: xmin: 178605 ymin: 329714 xmax: 181390 ymax: 333611

Projected CRS: Amersfoort / RD New

# A tibble: 155 × 13

	cadmium	copper	lead	zinc	elev	dist	om	ffreq				
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<fct>				
1	11.7	85	299	1022	7.91	0.00136	13.6	1				
2	8.6	81	277	1141	6.98	0.0122	14	1				
3	6.5	68	199	640	7.8	0.103	13	1				
4	2.6	81	116	257	7.66	0.190	8	1				
5	2.8	48	117	269	7.48	0.277	8.7	1				
6	3	61	137	281	7.79	0.364	7.8	1				
7	3.2	31	132	346	8.22	0.190	9.2	1				
8	2.8	29	150	406	8.49	0.0922	9.5	1				
9	2.4	37	133	347	8.67	0.185	10.6	1				
10	1.6	24	80	183	9.05	0.310	6.3	1				
# i 145 more rows												
# i 5 more variables: soil <fct>, lime <fct>,												
# landuse <fct>, dist.m <dbl>, geometry<POINT [n]t>												

```
1 ( meuse_riv = st_polygon(list(meuse.riv)) |>
2   st_sfc() |>
3   st_set_crs(28992) |>
4   st_as_sf()
5 )
```

Simple feature collection with 1 feature and 0 fields

Geometry type: POLYGON

Dimension: XY

Bounding box: xmin: 178304 ymin: 325698.5 xmax: 182331.5 ymax: 337684.8

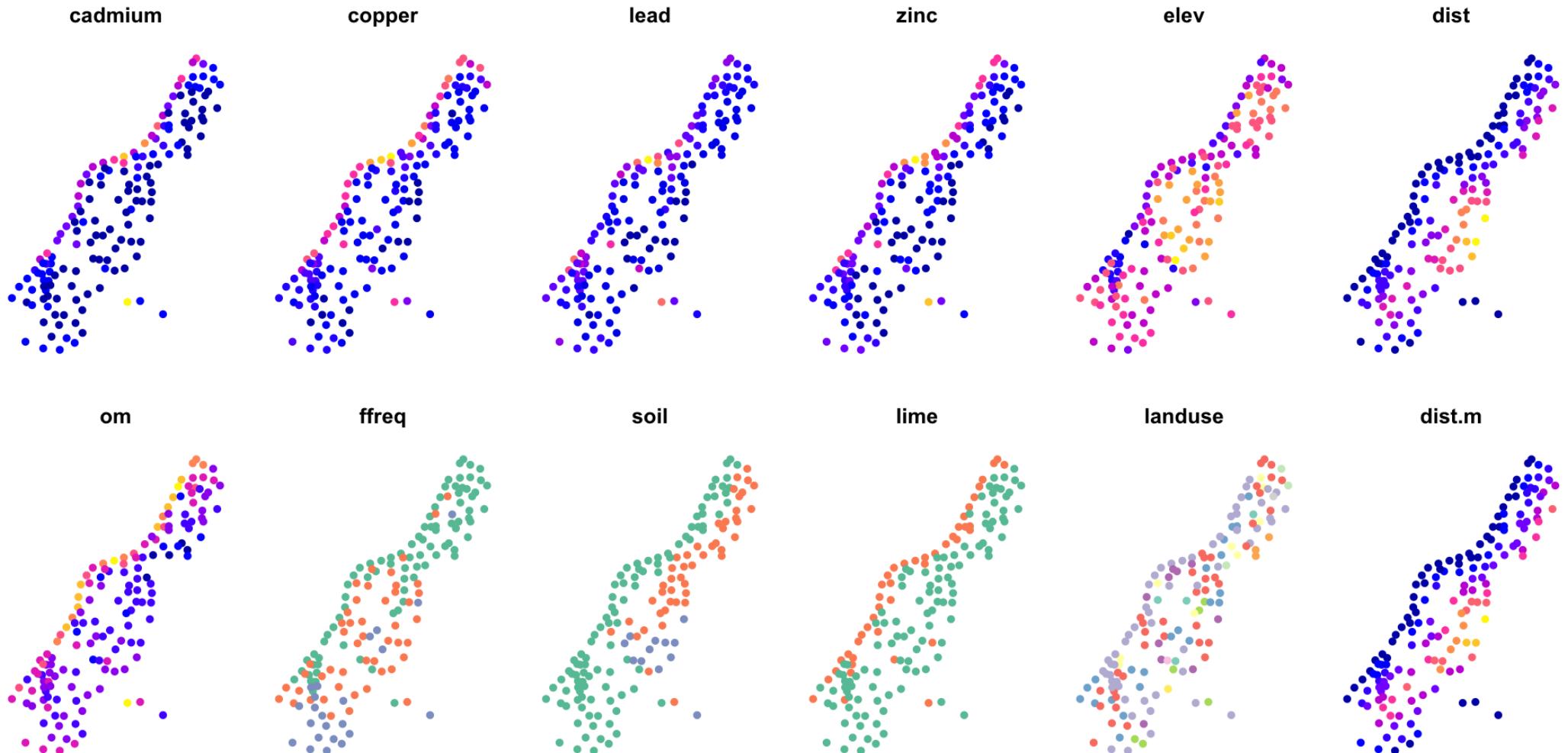
Projected CRS: Amersfoort / RD New

x

1 POLYGON ((182003.7 337678.6...

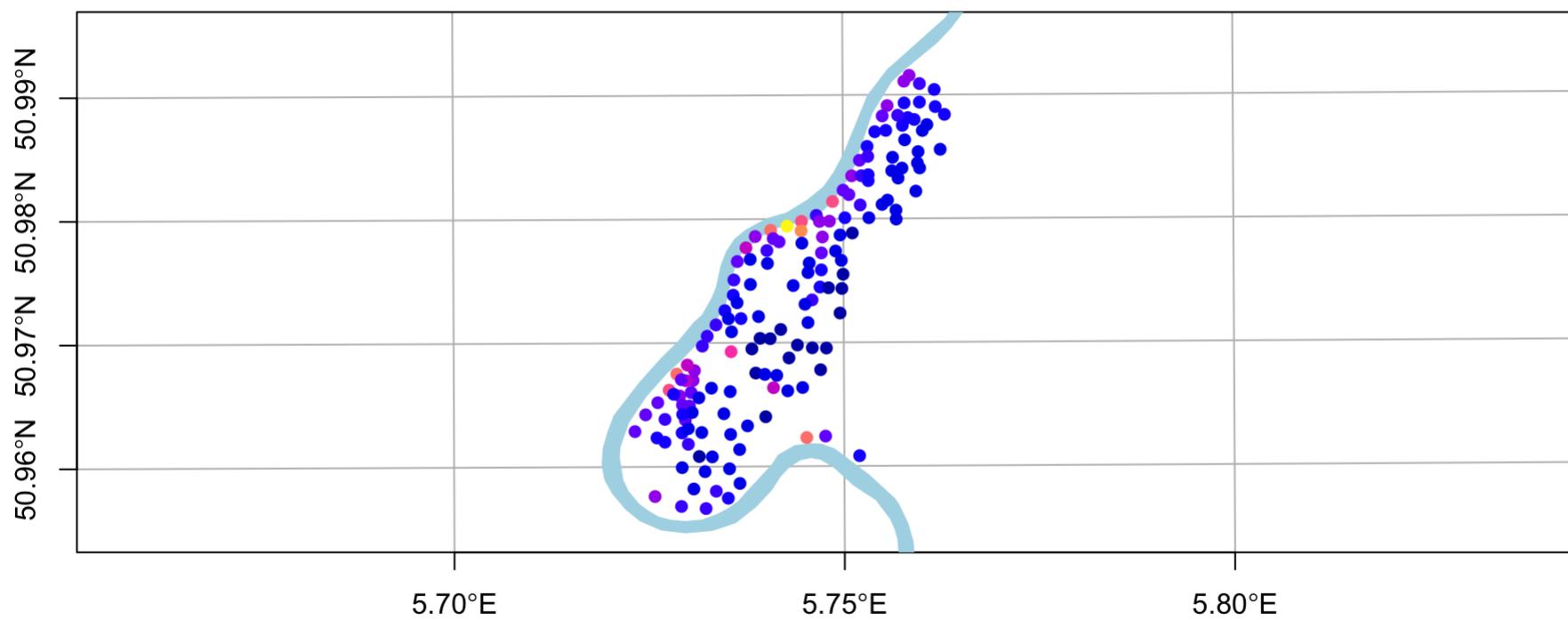
# Meuse

```
1 plot(meuse, pch=16, max.plot=12)
```



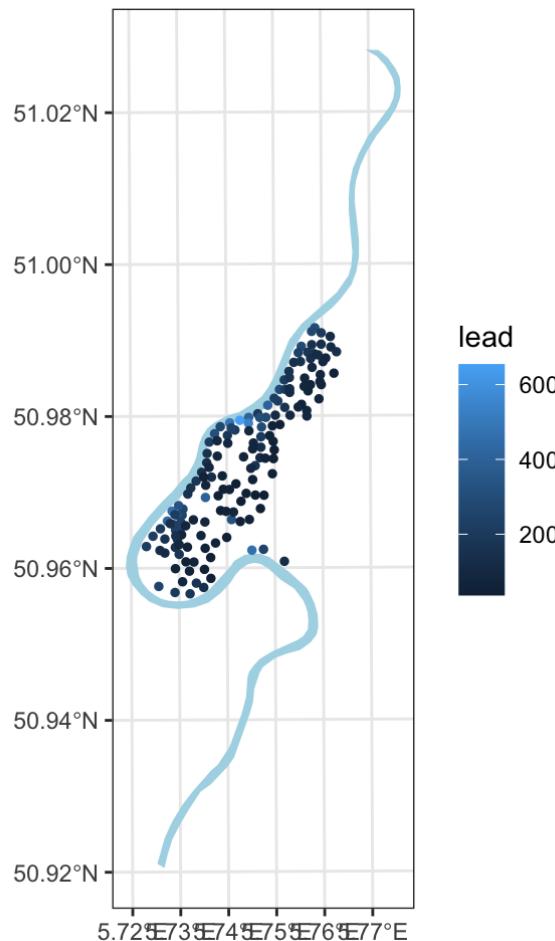
# Layering plots

```
1 plot(  
2   meuse_riv, col=adjustcolor("lightblue", alpha.f=1), border = NA,  
3   axes=TRUE, graticule=st_crs(4326),  
4   ylim = c(329500, 334000)  
5 )  
6 plot(meuse[, "lead"], pch=16, add=TRUE)
```



# ggplot2

```
1 ggplot() +  
2   geom_sf(data=meuse_riv, fill="lightblue", color=NA) +  
3   geom_sf(data=meuse, aes(color=lead), size=1)
```



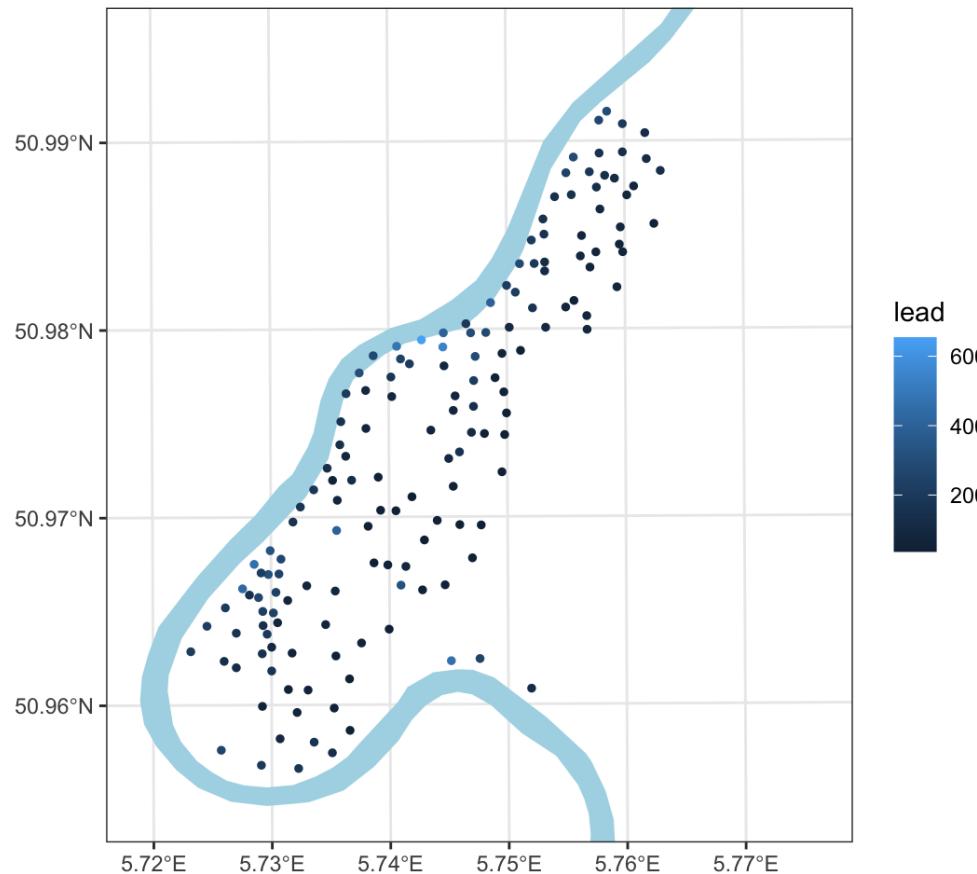
# ggplot2 - axis limits

```
1 ggplot() +  
2   geom_sf(data=meuse_riv, fill="lightblue", color=NA) +  
3   geom_sf(data=meuse, aes(color=lead), size=1) +  
4   ylim(50.95, 50.99)
```



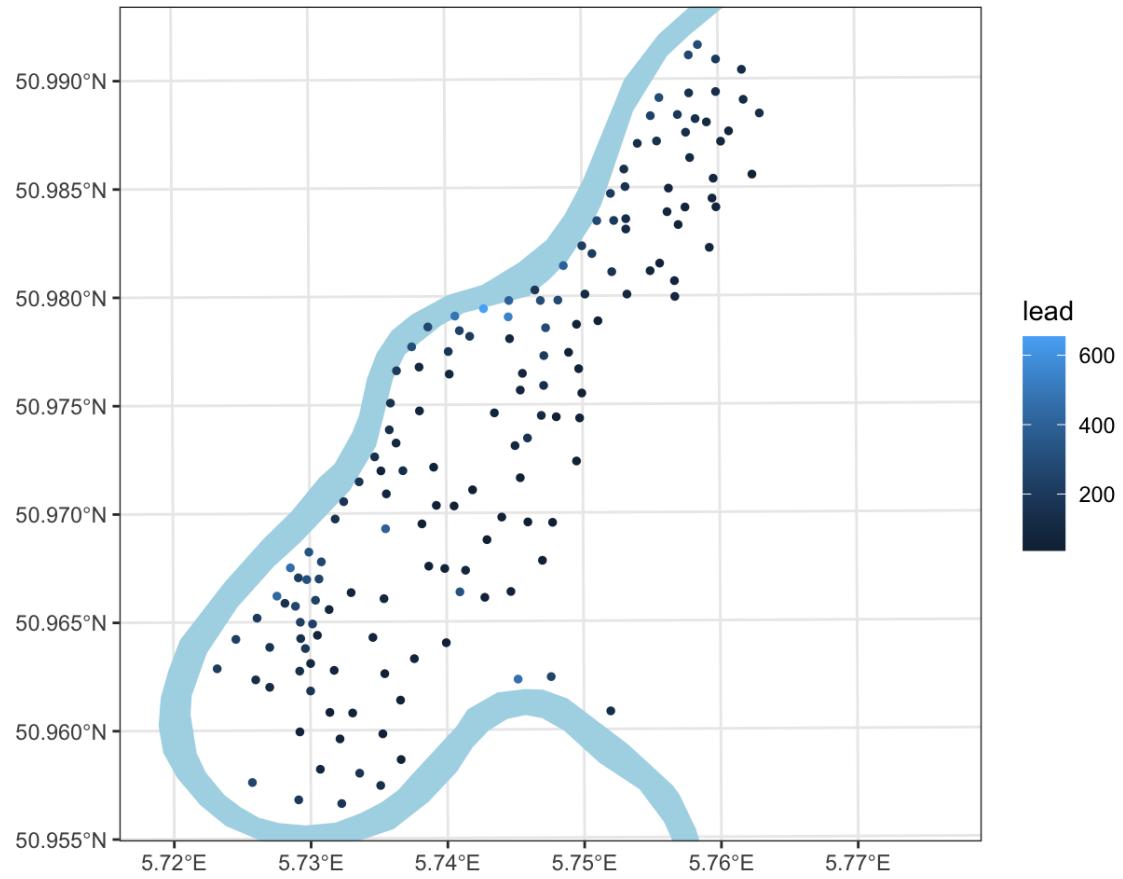
# ggplot2 - axis limits

```
1 ggplot() +  
2   geom_sf(data=meuse_riv, fill="lightblue", color=NA) +  
3   geom_sf(data=meuse, aes(color=lead), size=1) +  
4   ylim(329500, 334000)
```



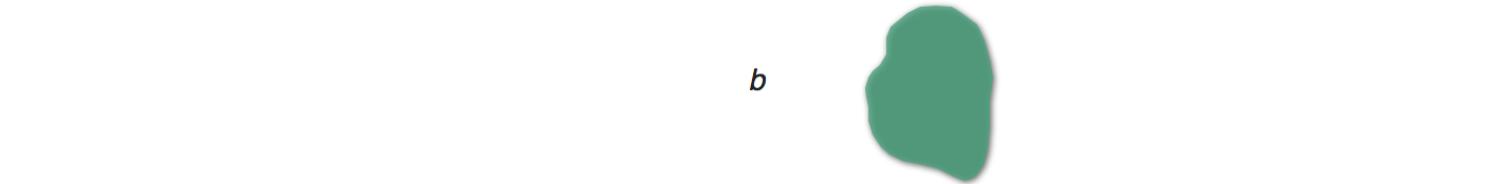
# ggplot2 - bounding box

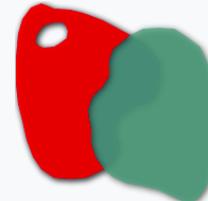
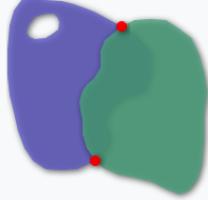
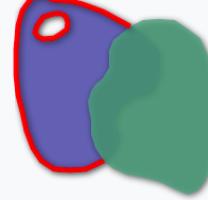
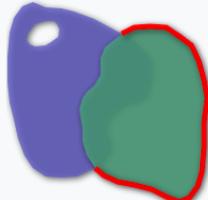
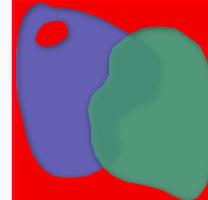
```
1 ggplot() +  
2   geom_sf(data=st_sf(meuse_riv), fill="lightblue", color=NA) +  
3   geom_sf(data=meuse, aes(color=lead), size=1) +  
4   ylim(st_bbox(meuse)[ "ymin" ], st_bbox(meuse)[ "ymax" ])
```



# Geometry Predicates

# DE-9IM



	<b>Interior</b>	<b>Boundary</b>	<b>Exterior</b>
<b>Interior</b>			
<b>Boundary</b>			
<b>Exterior</b>			

$\dim[I(a) \cap I(b)] = 2$      $\dim[I(a) \cap B(b)] = 1$      $\dim[I(a) \cap E(b)] = 2$   
 $\dim[B(a) \cap I(b)] = 1$      $\dim[B(a) \cap B(b)] = 0$      $\dim[B(a) \cap E(b)] = 1$   
 $\dim[E(a) \cap I(b)] = 2$      $\dim[E(a) \cap B(b)] = 1$      $\dim[E(a) \cap E(b)] = 2$

# Spatial predicates

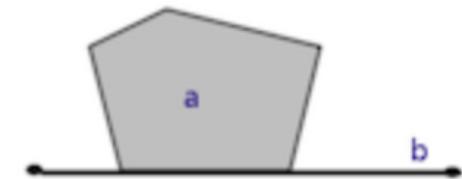
Within(a,b)



Touches(a,b)



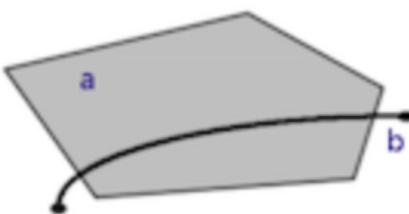
Touches(a,b)



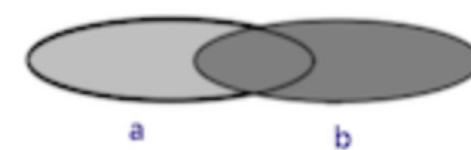
Crosses(a,b)



Crosses(a,b)



Overlaps(a,b)



st\_within(a,b):

```
\[ \begin{bmatrix} T & * & F \\ * & * & F \\ * & * & * \end{bmatrix} \]
```

st\_touches(a,b):

```
\[ \begin{bmatrix} F & T & * \\ T & * & * \\ * & * & * \end{bmatrix} \cup \begin{bmatrix} F & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \cup \begin{bmatrix} F & * & * \\ * & * & T \\ * & * & * \end{bmatrix} \]
```

# Sparse vs Full Results

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

```
List of 11
$ : 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
- attr(*, "class")= chr [1:2] "sgbp" "list"
```

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

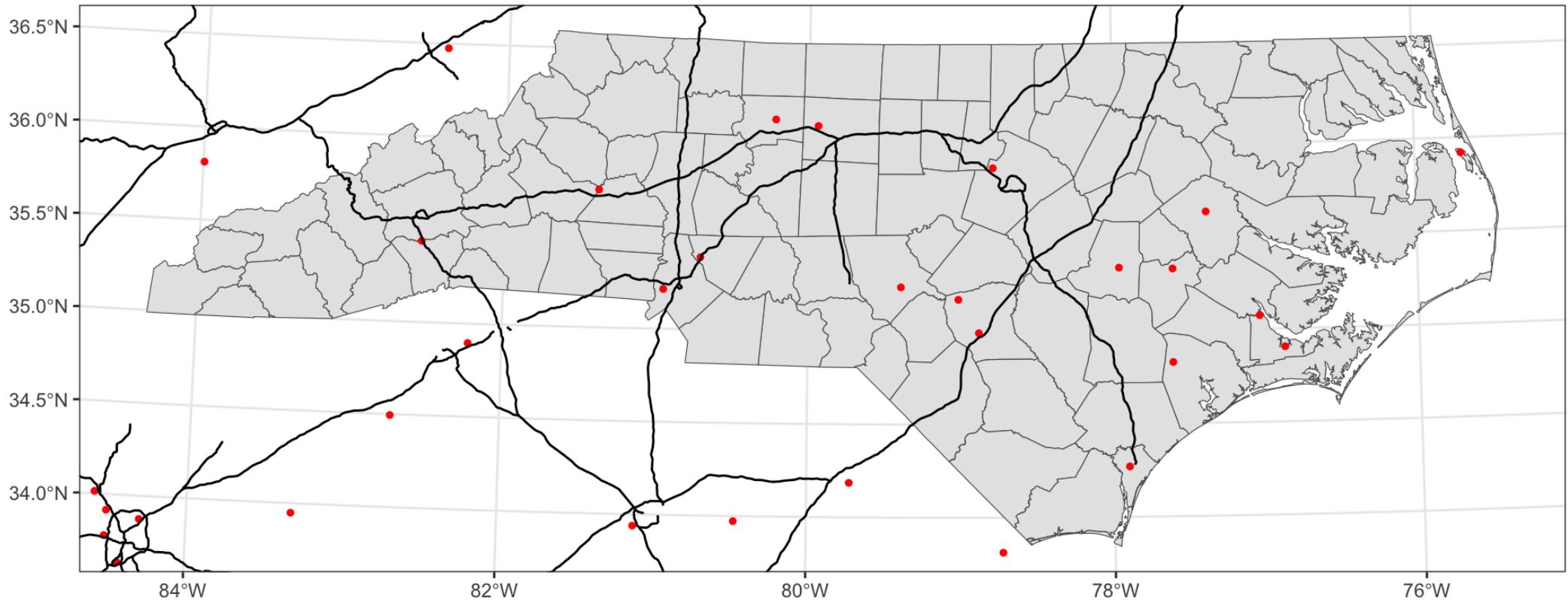
```
logi [1:100, 1:940] FALSE FALSE FALSE FALSE FALSE FALSE ...
```

# Examples

- Which counties have an airport?
- Which counties are adjacent to Durham County?
- Which counties have more than 4 neighbors?

```
1 ncc = read_sf("data/gis/nc_counties/", quiet=TRUE) |> st_transform(3631)
2 air = read_sf("data/gis/airports/", quiet=TRUE) |> st_transform(3631)
3 hwy = read_sf("data/gis/us_interstates/", quiet=TRUE) |> st_transform(3631)
```

# Data



# Which counties have an airport?

```
1 ncc |>
2   select(COUNTY, geometry) |>
3   mutate(
4     airports = st_intersects(ncc, air) |> unclass(),
5     n = purrr::map_int(airports, length),
6     airport_names = purrr::map_chr(
7       airports,
8       ~ paste(air$AIRPT_NAME[.x], collapse=", "))
9     ) |>
10    filter(n > 0) |>
11    arrange(desc(n))
```

Simple feature collection with 16 features and 4 fields

Geometry type: MULTIPOLYGON

Dimension: XY

Bounding box: xmin: 257742.1 ymin: 20311.68 xmax: 930352.8 ymax: 280007

Projected CRS: NAD83(NSRS2007) / North Carolina

# A tibble: 16 × 5

	COUNTY	geometry	airports	n	airport_names
	<chr>	<MULTIPOLYGON [m]>	<list>	<int>	<chr>
1	Craven County	(((815892.7 133083, 815712 132990.1, 8...	<int>	2	Cherry Point...
2	Cumberland County	(((634732.8 168173.4, 634781.1 168150...	<int>	2	Pope Air For...
3	Forsyth County	(((480355.6 279558.3, 480622.3 279554...	<int>	1	Smith Reynol...
4	Guilford County	(((516951.1 278659.3, 517346.4 278644...	<int>	1	Piedmont Tri...

Sta 344/644 - Fall 2023

5	Dare County	((925331.7 195868.4, 925150.9 195670... <int>	1	Dare County ...
6	Wake County	((635486.1 258098.2, 635729.3 258098... <int>	1	Raleigh-Durh...
7	Pitt County	((756611.7 231604, 757952.2 231061.2,... <int>	1	Pitt-Greenvi...
8	Catawba County	((416107.9 232317.8, 416052.5 231678... <int>	1	Hickory Regi...
9	Buncombe County	((304720 234673.2, 304825.7 234591.9,... <int>	1	Asheville Re...
10	Wayne County	((700246.7 204130.8, 700364.9 204087... <int>	1	Seymour John...
11	Mecklenburg County	((446654.6 196384.3, 446836 196381, 4... <int>	1	Charlotte/Do...
12	Moore County	((577854 196164.2, 577890.8 195888.4,... <int>	1	Moore County...
13	Cabarrus County	((453375.4 196276.2, 453511.2 196262... <int>	1	Concord Regi...

# Which counties neighbor Durham County?

```
1 ncc |>
2   select(COUNTY, geometry) |>
3   mutate(
4     touch_durham = st_touches(ncc, ncc) |> filter(COUNTY == "Durham County")) |> unclass(),
5     n_touches = map_int(touch_durham, length)
6   ) |>
7   filter(n_touches > 0)
```

Simple feature collection with 5 features and 3 fields

Geometry type: MULTIPOLYGON

Dimension: XY

Bounding box: xmin: 559195.8 ymin: 195938.7 xmax: 676918.6 ymax: 309925.7

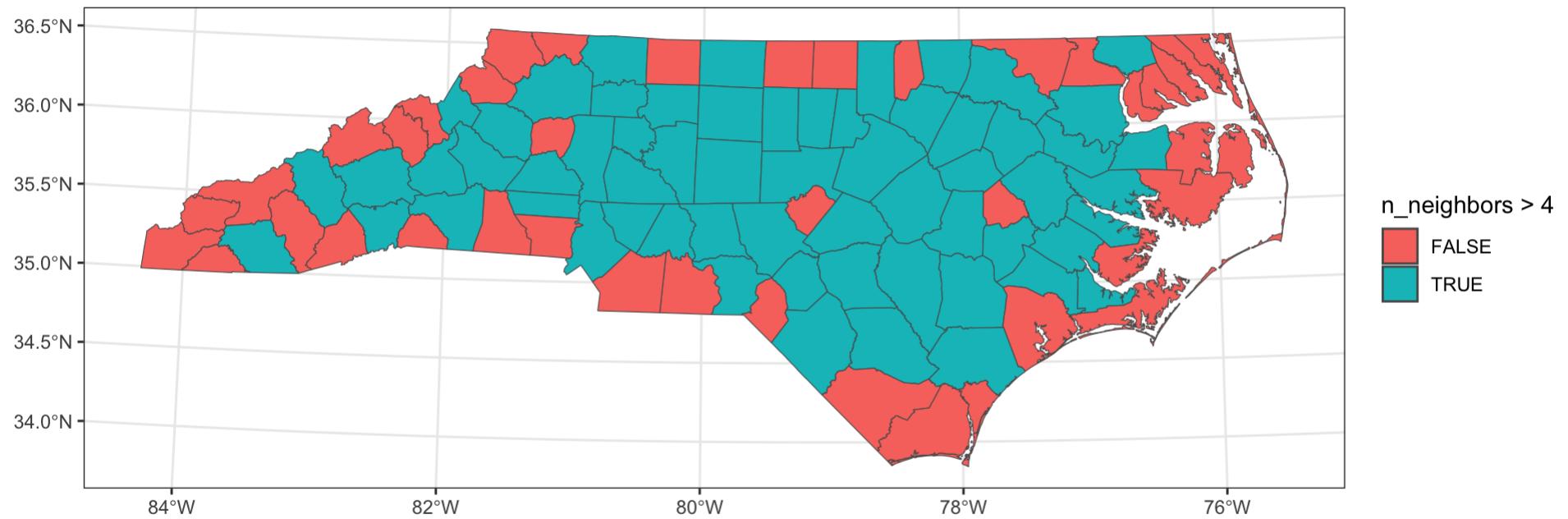
Projected CRS: NAD83(NSRS2007) / North Carolina

# A tibble: 5 × 4

	COUNTY	geometry	touch_durham	n_touches
*	<chr>	<MULTIPOLYGON [m]>	<list>	<int>
1	Person County	(((611470.1 309761.9, 611935.8 309762, 612123.9..., 611470.1 309761.9))	<int [1]>	1
2	Granville County	(((658298.4 309773.1, 658351.6 309762.4, 658405..., 658298.4 309773.1))	<int [1]>	1
3	Orange County	(((586623.3 276687.6, 587090.7 276675.3, 587378..., 586623.3 276687.6))	<int [1]>	1
4	Wake County	(((635486.1 258098.2, 635729.3 258098.9, 635815..., 635486.1 258098.2))	<int [1]>	1
5	Chatham County	(((589371.3 235671.1, 589560.8 235648.5, 589606..., 589371.3 235671.1))	<int [1]>	1

# Which counties have more than 4 neighbors?

```
1 ncc |>
2   mutate(
3     neighbors = st_touches(ncc) |> unclass(),
4     n_neighbors = map_int(neighbors, length)
5   ) |>
6   ggplot(aes(fill = n_neighbors > 4)) +
7   geom_sf()
```



# Geometry Manipulation

# Casting

```
1 (nc_pts = st_cast(nc, "MULTIPOINT"))
```

Simple feature collection with 100 features and 7 fields

Geometry type: MULTIPOINT

Dimension: XY

Bounding box: xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax: 36.58965

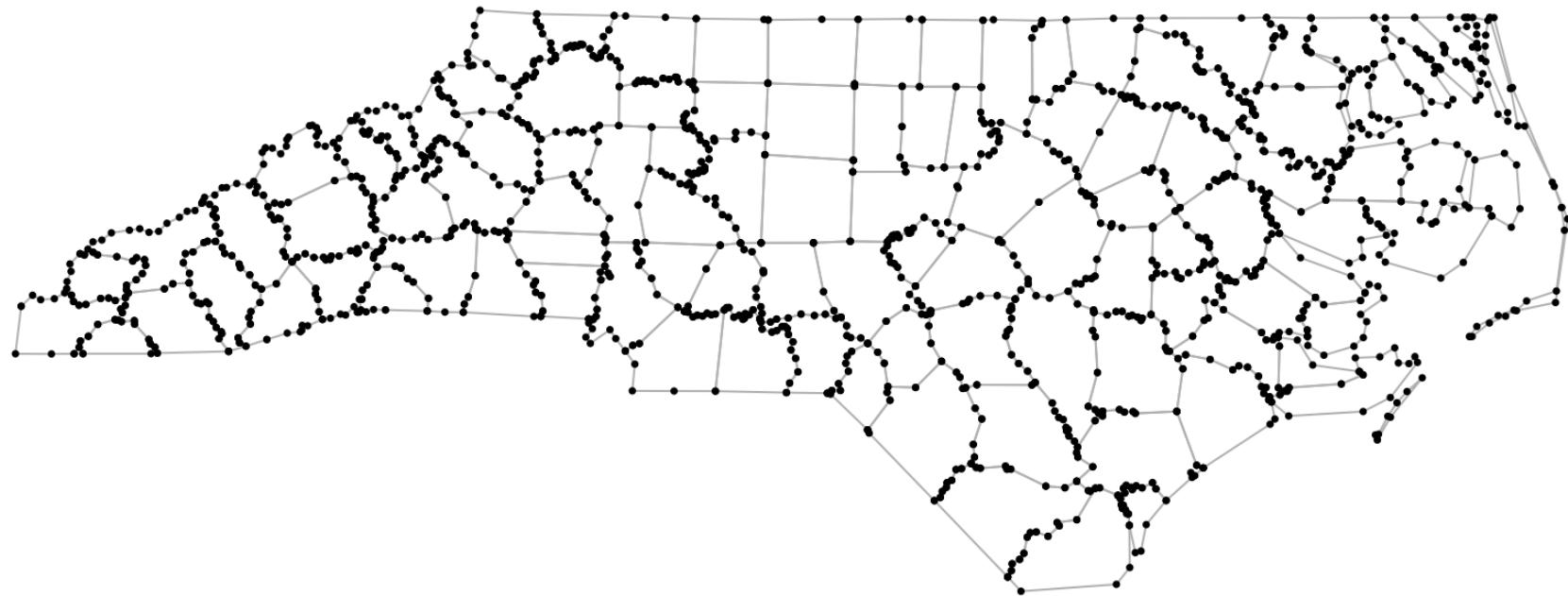
Geodetic CRS: NAD27

# A tibble: 100 × 8

	NAME	BIR74	SID74	NWBIR74	BIR79	SID79	NWBIR79	geometry
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<MULTIPOINT [°]>
1	Ashe	1091	1	10	1364	0	19	((-81.47276 36.23436), (-81.54084 ...
2	Alleghany	487	0	10	542	3	12	((-81.23989 36.36536), (-81.24069 ...
3	Surry	3188	5	208	3616	6	260	((-80.45634 36.24256), (-80.47639 ...
4	Currituck	508	1	123	830	2	145	((-76.00897 36.3196), (-76.01735 3...
5	Northhampton	1421	9	1066	1606	3	1197	((-77.21767 36.24098), (-77.23461 ...
6	Hertford	1452	7	954	1838	5	1237	((-76.74506 36.23392), (-76.98069 ...
7	Camden	286	0	115	350	2	139	((-76.00897 36.3196), (-75.95718 3...
8	Gates	420	0	254	594	2	371	((-76.56251 36.34057), (-76.60424 ...
9	Warren	968	4	748	1190	2	844	((-78.30876 36.26004), (-78.28293 ...
10	Stokes	1612	1	160	2038	5	176	((-80.02567 36.25023), (-80.45301 ...

# i 90 more rows

```
1 plot(st_geometry(nc), border='grey')
2 plot(st_geometry(nc_pts), pch=16, cex=0.5, add=TRUE)
```

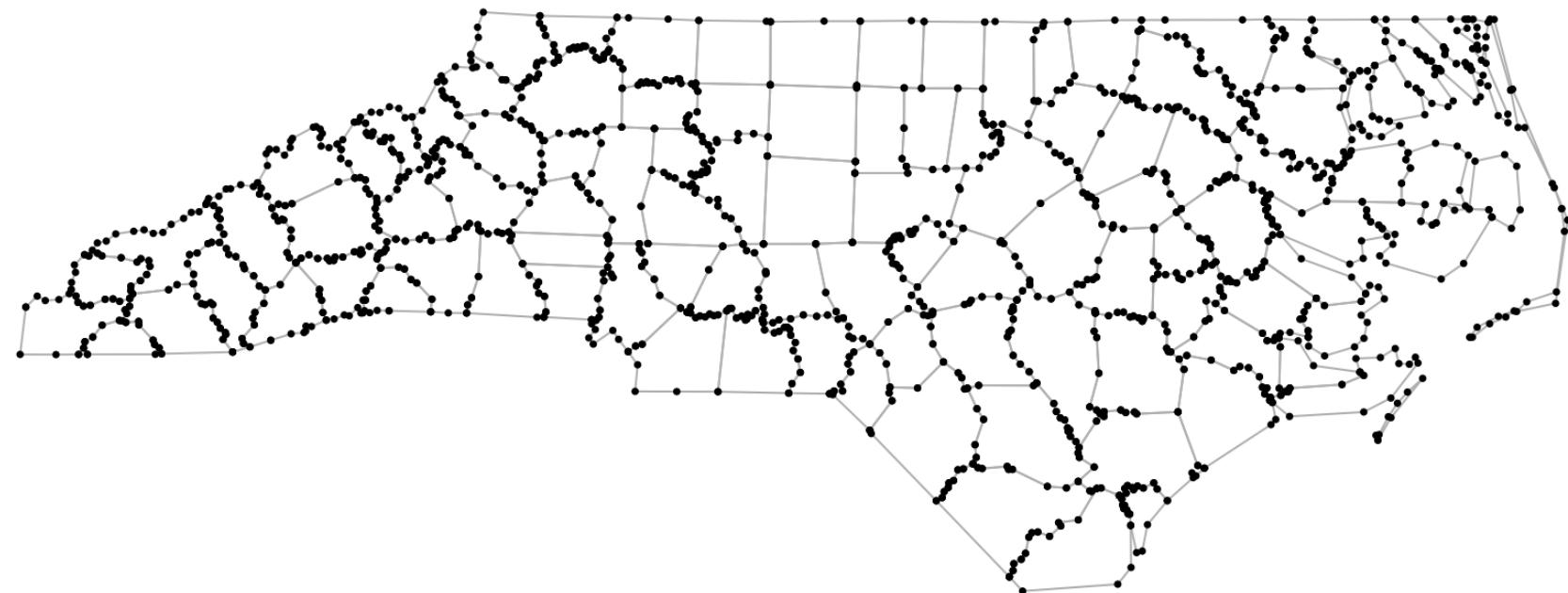


# Casting - POINT

```
1 st_cast(nc, "POINT")
```

```
Simple feature collection with 2529 features and 7 fields
Geometry type: POINT
Dimension:      XY
Bounding box:   xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax: 36.58965
Geodetic CRS:   NAD27
# A tibble: 2,529 × 8
  NAME    BIR74  SID74  NWBIR74  BIR79  SID79  NWBIR79       geometry
  <chr> <dbl>  <dbl>    <dbl>  <dbl>  <dbl>    <dbl>    <POINT [°]>
1 Ashe     1091     1        10    1364     0        19 (-81.47276 36.23436)
2 Ashe     1091     1        10    1364     0        19 (-81.54084 36.27251)
3 Ashe     1091     1        10    1364     0        19 (-81.56198 36.27359)
4 Ashe     1091     1        10    1364     0        19 (-81.63306 36.34069)
5 Ashe     1091     1        10    1364     0        19 (-81.74107 36.39178)
6 Ashe     1091     1        10    1364     0        19 (-81.69828 36.47178)
7 Ashe     1091     1        10    1364     0        19 (-81.7028 36.51934)
8 Ashe     1091     1        10    1364     0        19 (-81.67 36.58965)
9 Ashe     1091     1        10    1364     0        19 (-81.3453 36.57286)
10 Ashe    1091     1        10    1364     0        19 (-81.34754 36.53791)
# i 2,519 more rows
```

```
1 plot(st_geometry(nc), border='grey')
2 plot(st_geometry(st_cast(nc, "POINT"))), pch=16, cex=0.5, add=TRUE)
```



# Casting - LINESTRING

```
1 st_cast(nc, "MULTILINESTRING")
```

Simple feature collection with 100 features and 7 fields

Geometry type: MULTILINESTRING

Dimension: XY

Bounding box: xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax: 36.58965

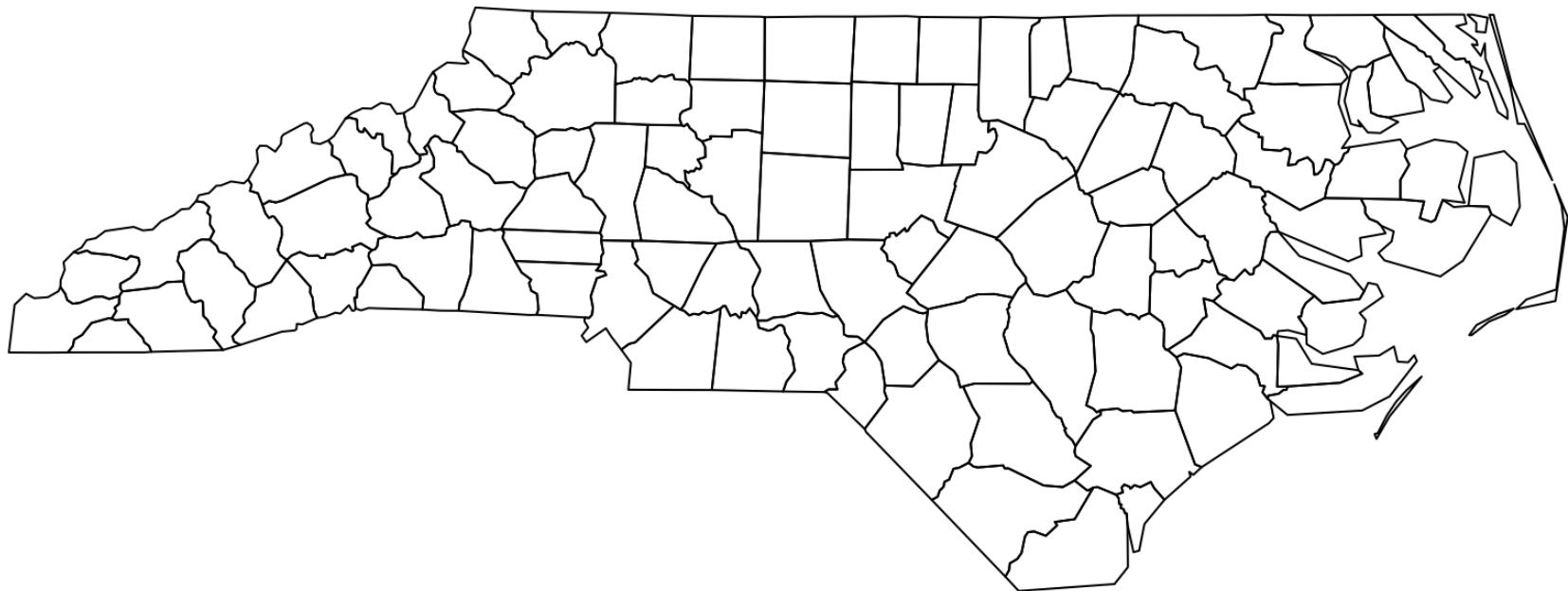
Geodetic CRS: NAD27

# A tibble: 100 × 8

	NAME	BIR74	SID74	NWBIR74	BIR79	SID79	NWBIR79	geometry
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<MULTILINESTRING [°]>
1	Ashe	1091	1	10	1364	0	19	((-81.47276 36.23436, -81.54084 36...
2	Alleghany	487	0	10	542	3	12	((-81.23989 36.36536, -81.24069 36...
3	Surry	3188	5	208	3616	6	260	((-80.45634 36.24256, -80.47639 36...
4	Currituck	508	1	123	830	2	145	((-76.00897 36.3196, -76.01735 36...
5	Northhampton	1421	9	1066	1606	3	1197	((-77.21767 36.24098, -77.23461 36...
6	Hertford	1452	7	954	1838	5	1237	((-76.74506 36.23392, -76.98069 36...
7	Camden	286	0	115	350	2	139	((-76.00897 36.3196, -75.95718 36...
8	Gates	420	0	254	594	2	371	((-76.56251 36.34057, -76.60424 36...
9	Warren	968	4	748	1190	2	844	((-78.30876 36.26004, -78.28293 36...
10	Stokes	1612	1	160	2038	5	176	((-80.02567 36.25023, -80.45301 36...

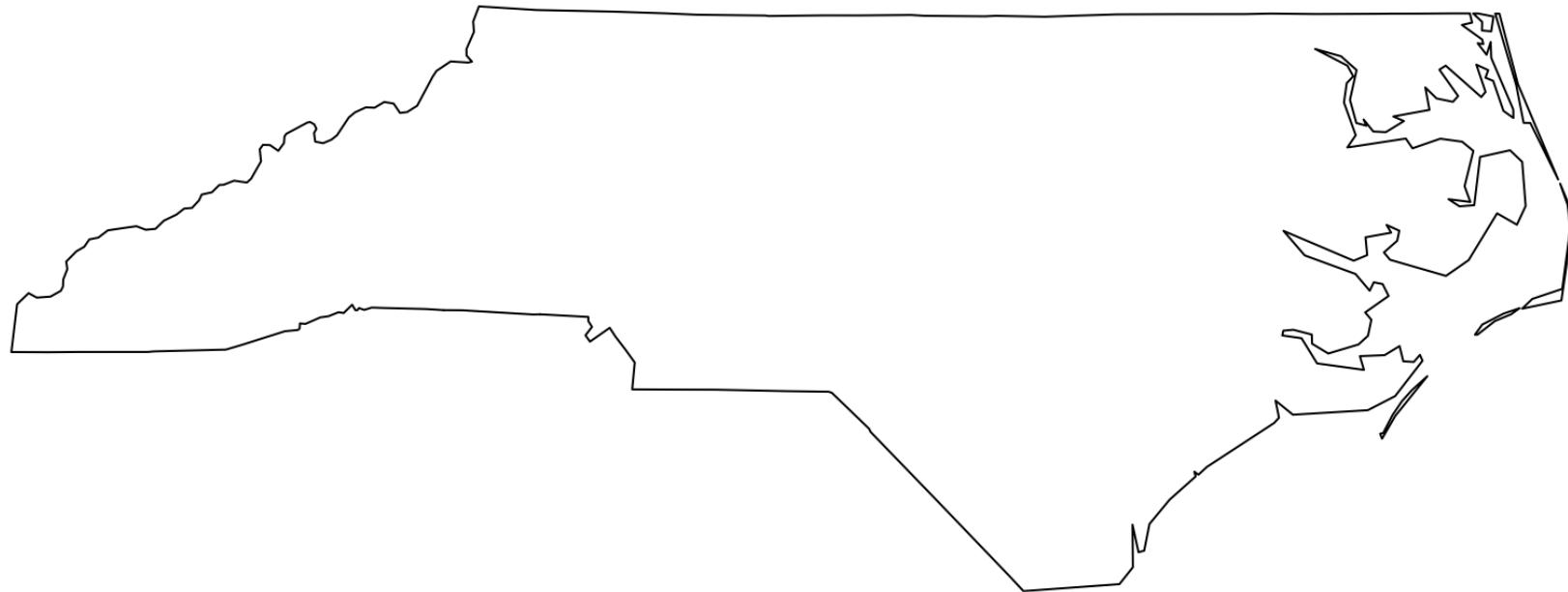
# i 90 more rows

```
1 st_cast(nc, "MULTILINESTRING") |> st_geometry() |> plot()
```



# Grouping Features

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

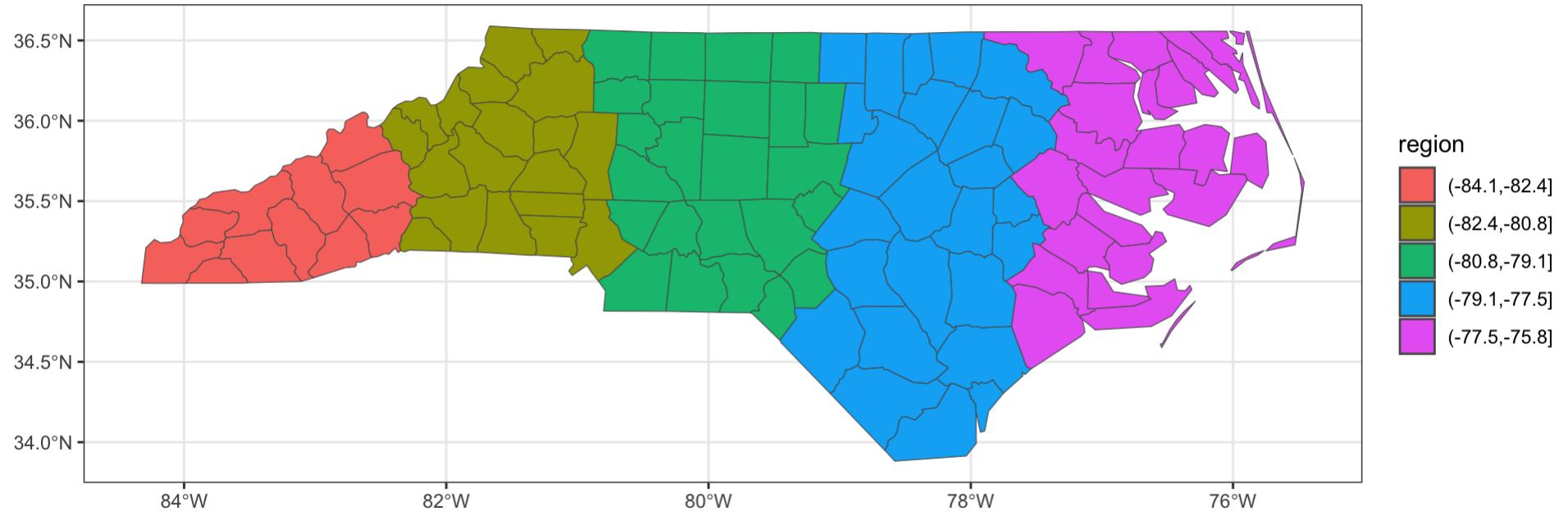


# More Grouping

```
1 ( nc_cut = nc |>
  2   mutate(X = st_centroid(nc) |> st_coordinates() |> (\(x) x[,1]))() |>
  3   mutate(region = cut(X, breaks = 5)) )
```

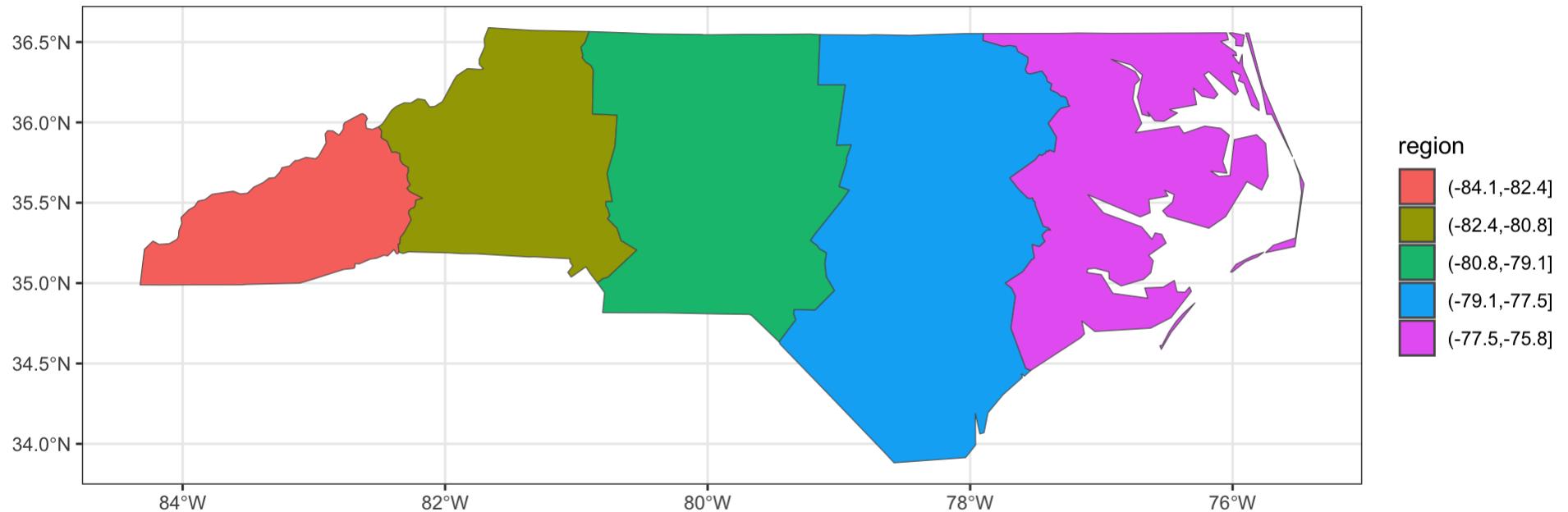
```
Simple feature collection with 100 features and 9 fields
Geometry type: MULTIPOLYGON
Dimension:      XY
Bounding box:   xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax: 36.58965
Geodetic CRS:   NAD27
# A tibble: 100 × 10
  NAME    BIR74  SID74  NWBIR74  BIR79  SID79  NWBIR79      geometry     x region
* <chr>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <MULTIPOLYGON [°]> <dbl> <fct>
  1 Ashe     1091     1      10    1364     0      19 (((-81.47276 36.23436, ...
  2 Allegha... 487      0      10     542     3      12 (((-81.23989 36.36536, ...
  3 Surry    3188      5     208    3616     6      260 (((-80.45634 36.24256, ...
  4 Curritu... 508      1     123     830     2      145 (((-76.00897 36.3196, ...
  5 Northam... 1421      9    1066    1606     3      1197 (((-77.21767 36.24098, ...
  6 Hertford 1452      7     954    1838     5      1237 (((-76.74506 36.23392, ...
  7 Camden    286      0     115     350     2      139 (((-76.00897 36.3196, ...
  8 Gates     420      0     254     594     2      371 (((-76.56251 36.34057, ...
  9 Warren    968      4     748    1190     2      844 (((-78.30876 36.26004, ...
 10 Stokes   1612      1     160    2038     5      176 (((-80.02567 36.25023, ...
# i 90 more rows
```

```
1 ggplot(nc_cut) +  
2   geom_sf(aes(fill=region))
```



# Union via summarize

```
1 nc_cut |>
2   group_by(region) |>
3   summarize() |>
4   ggplot() +
5     geom_sf(aes(fill=region))
```



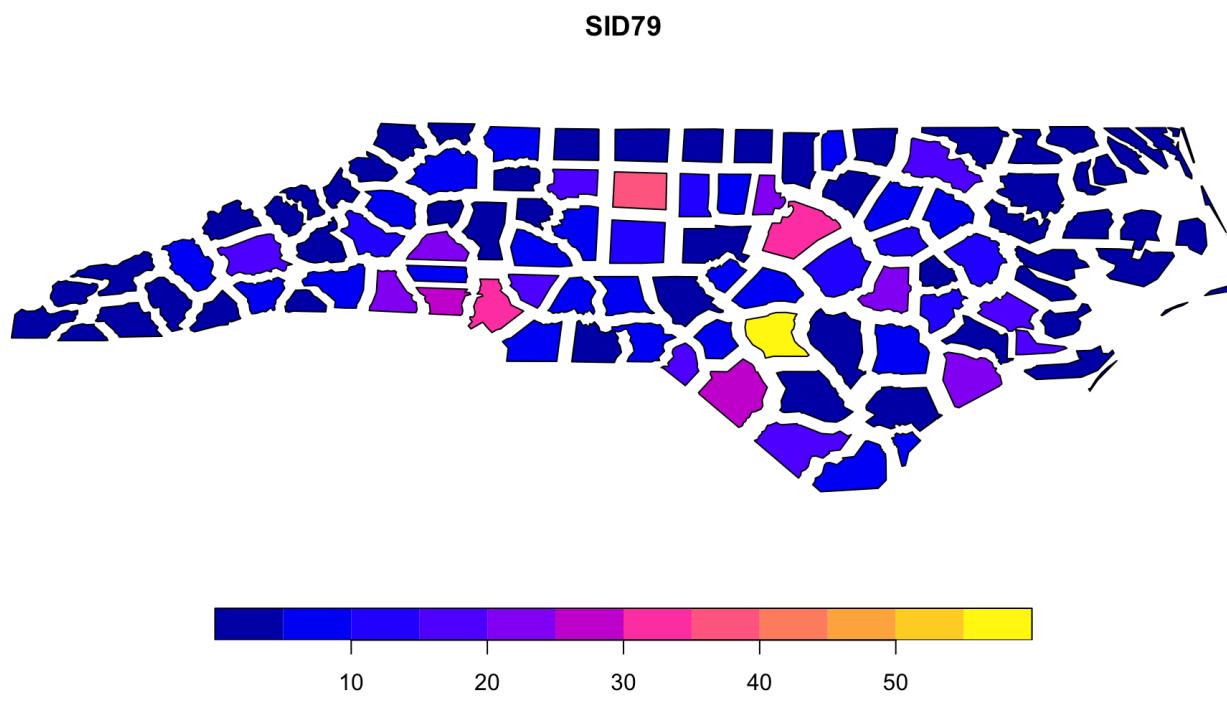
# Affine Transformations

```
1 rotate = function(a) matrix(c(cos(a), sin(a), -sin(a), cos(a)), 2, 2)
2
3 ctrd = st_centroid(nc_state)
4 state_rotate = (nc_state) * rotate(-pi/4)
5 plot(state_rotate, axes=TRUE)
```



# Scaling Size

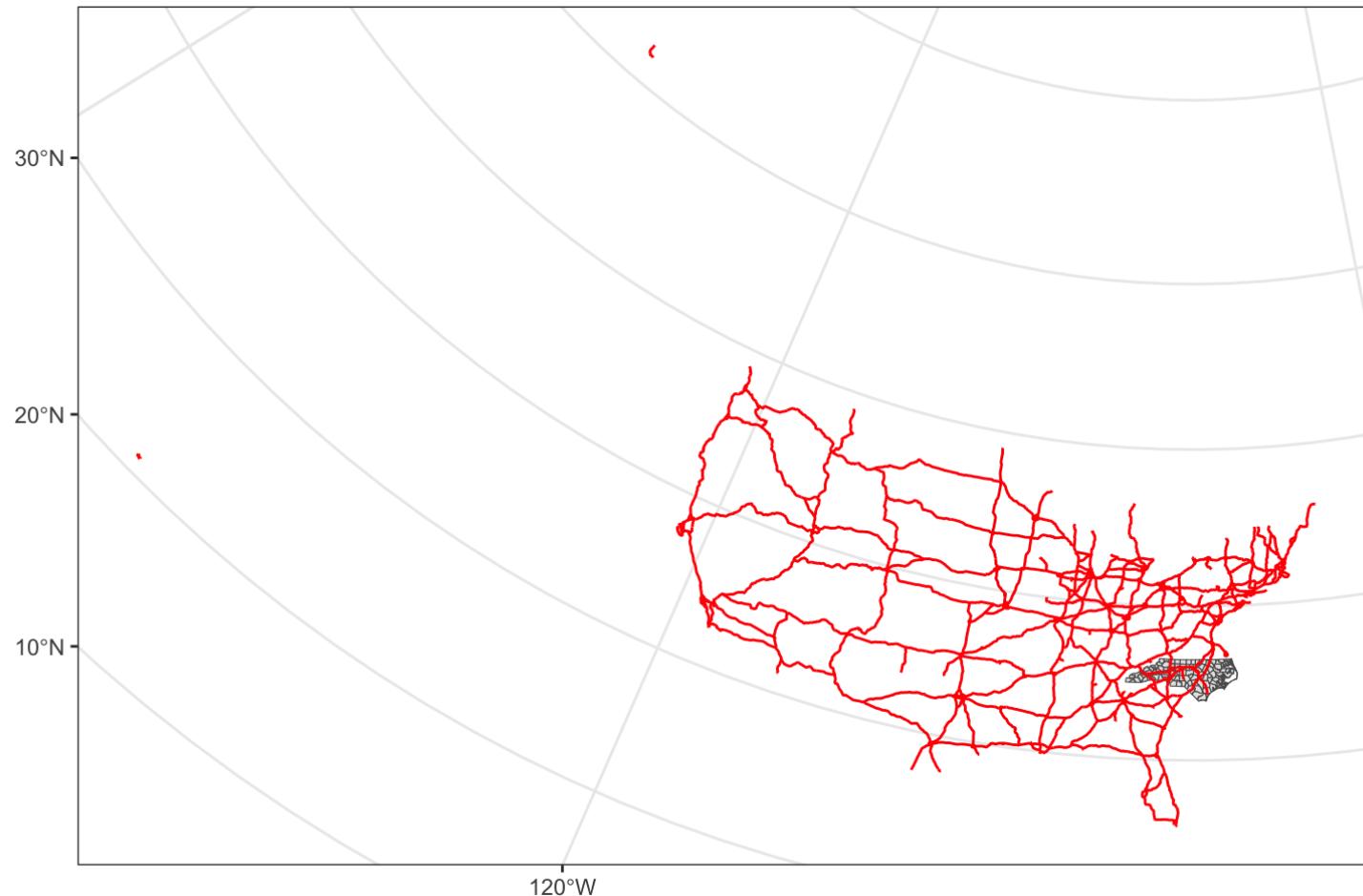
```
1 ctrd = st_centroid(st_geometry(nc))
2 area = st_area(nc) |> strip_attrs()
3
4 nc_rot = nc
5 st_geometry(nc_rot) = (st_geometry(nc) - ctrd) * 0.75 + ctrd
6
7 plot(nc_rot[, "SID79"])
```



# Highway Example

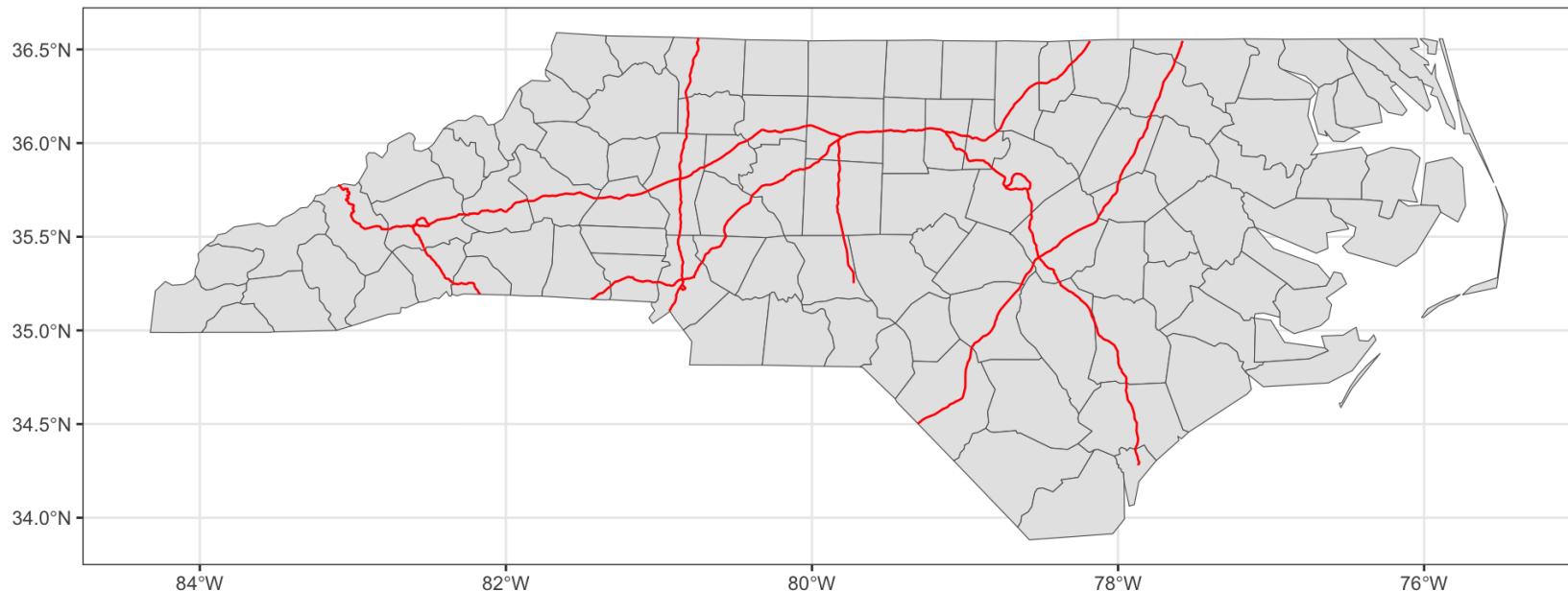
# Highways

```
1 ggplot() +  
2   geom_sf(data=ncc) +  
3   geom_sf(data=hwy, col='red')
```



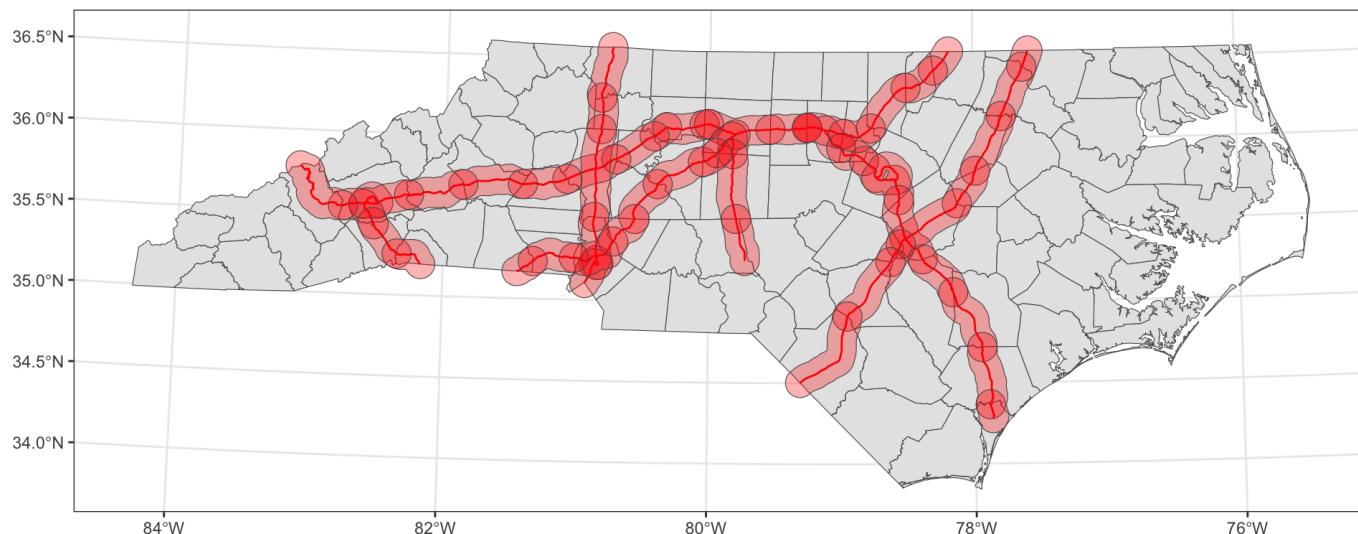
# NC Interstate Highways

```
1 hwy_nc = st_intersection(hwy, ncc)
2
3 ggplot() +
4   geom_sf(data=ncc) +
5   geom_sf(data=hwy_nc, col='red')
```



# Counties near the interstate (Buffering)

```
1 hwy_nc_buffer = hwy_nc |>  
2   st_buffer(10000)  
3  
4 ggplot() +  
5   geom_sf(data=ncc) +  
6   geom_sf(data=hwy_nc, color='red') +  
7   geom_sf(data=hwy_nc_buffer, fill='red', alpha=0.3)
```



# Counties near the interstate (Buffering + Union)

```
1 hwy_nc_buffer = hwy_nc |>  
2   st_buffer(10000) |>  
3   st_union() |>  
4   st_sf()
```

```
1 ggplot() +  
2   geom_sf(data=ncc) +  
3   geom_sf(data=hwy_nc, color='red') +  
4   geom_sf(data=hwy_nc_buffer, fill='red', alp
```

# Example

How many counties in North Carolina are within 5, 10, 20, or 50 km of an interstate highway?

```
1 hwy_nc |>
2   st_buffer(10000) |>
3   st_union() |>
4   st_intersects(ncc, y = _) |>
5   map_lgl(~ length(.x) >= 1) |>
6   sum()
```

```
[1] 55
```

# Gerrymandering Example

# NC House Districts - 112th Congress

```
1 nc_house = read_sf("data/nc_districts112.gpkg", quiet = TRUE) |>
2   select(ID, DISTRICT) |>
3   mutate(DISTRICT = as_factor(DISTRICT))
4 )
```

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

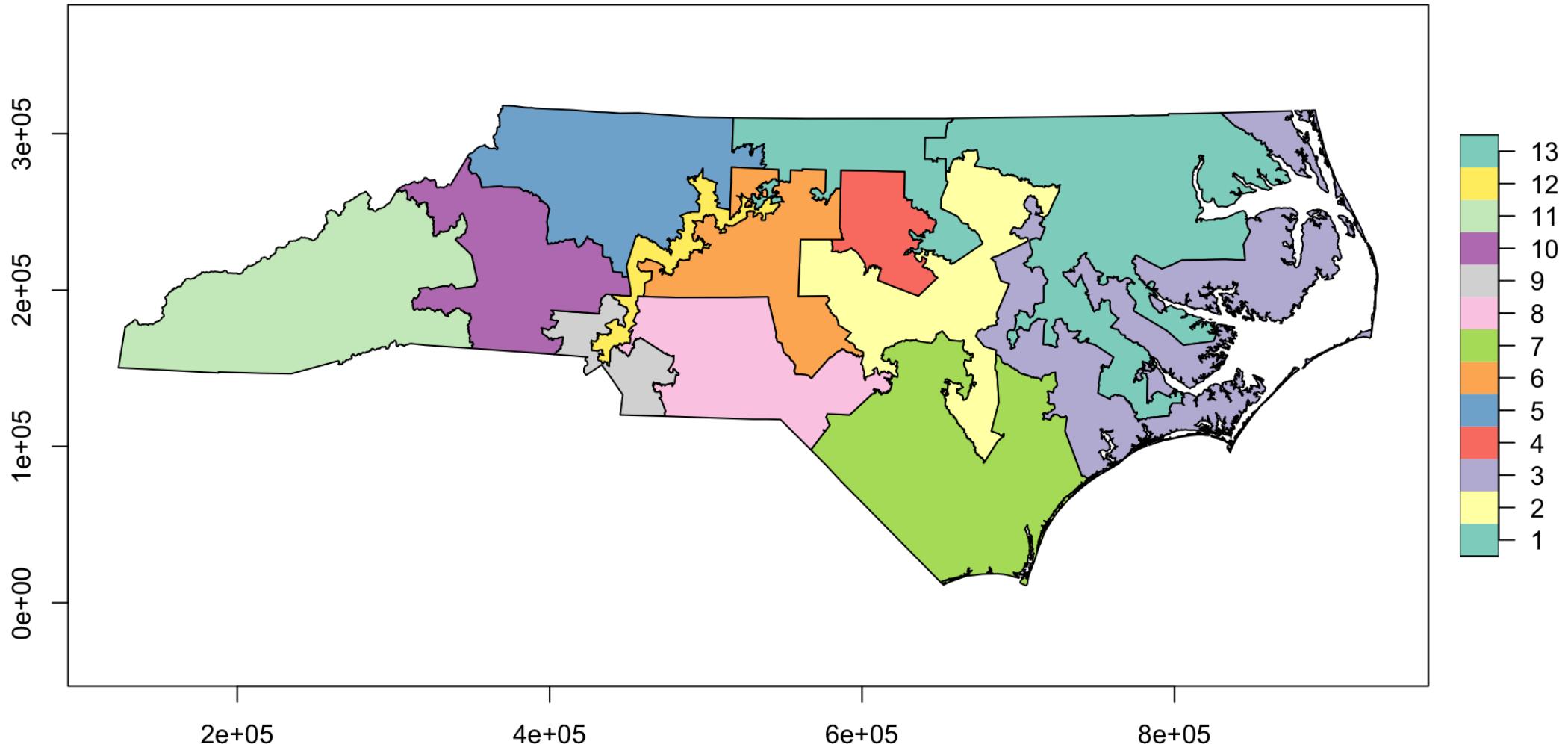
	ID	DISTRICT	geom
*	<chr>	<fct>	<MULTIPOLYGON [°]>
1	037108112001	1	(((-77.32845 35.35031, -77.35398 35.32799, -77.33...
2	037108112002	2	(((-78.89928 35.12619, -78.89763 35.12859, -78.89...
3	037108112003	3	(((-75.68266 35.23291, -75.68113 35.23237, -75.68...
4	037108112004	4	(((-78.77926 35.78568, -78.77947 35.77568, -78.79...
5	037108112005	5	(((-79.8968 36.38075, -79.89213 36.37108, -79.892...
6	037108112006	6	(((-80.4201 35.68953, -80.41483 35.68918, -80.411...
7	037108112007	7	(((-77.59169 34.40907, -77.58699 34.40611, -77.58...
8	037108112008	8	(((-78.93373 34.95909, -78.94074 34.95789, -78.94...

9 037108112009 9

(((-80.93058 35.18181, -80.9244 35.16754, -80.921...

```
1 nc_house = st_transform(nc_house, 3631)
2 plot(nc_house[, "DISTRICT"], axes=TRUE)
```

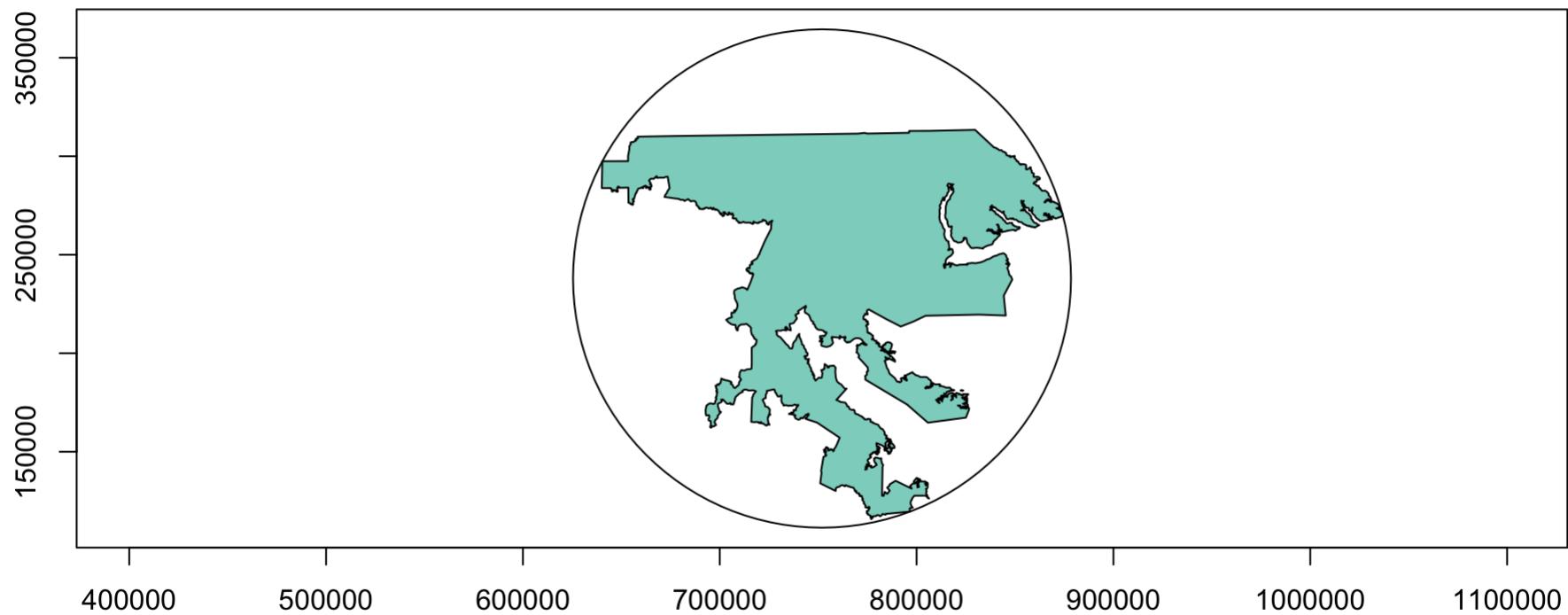
DISTRICT



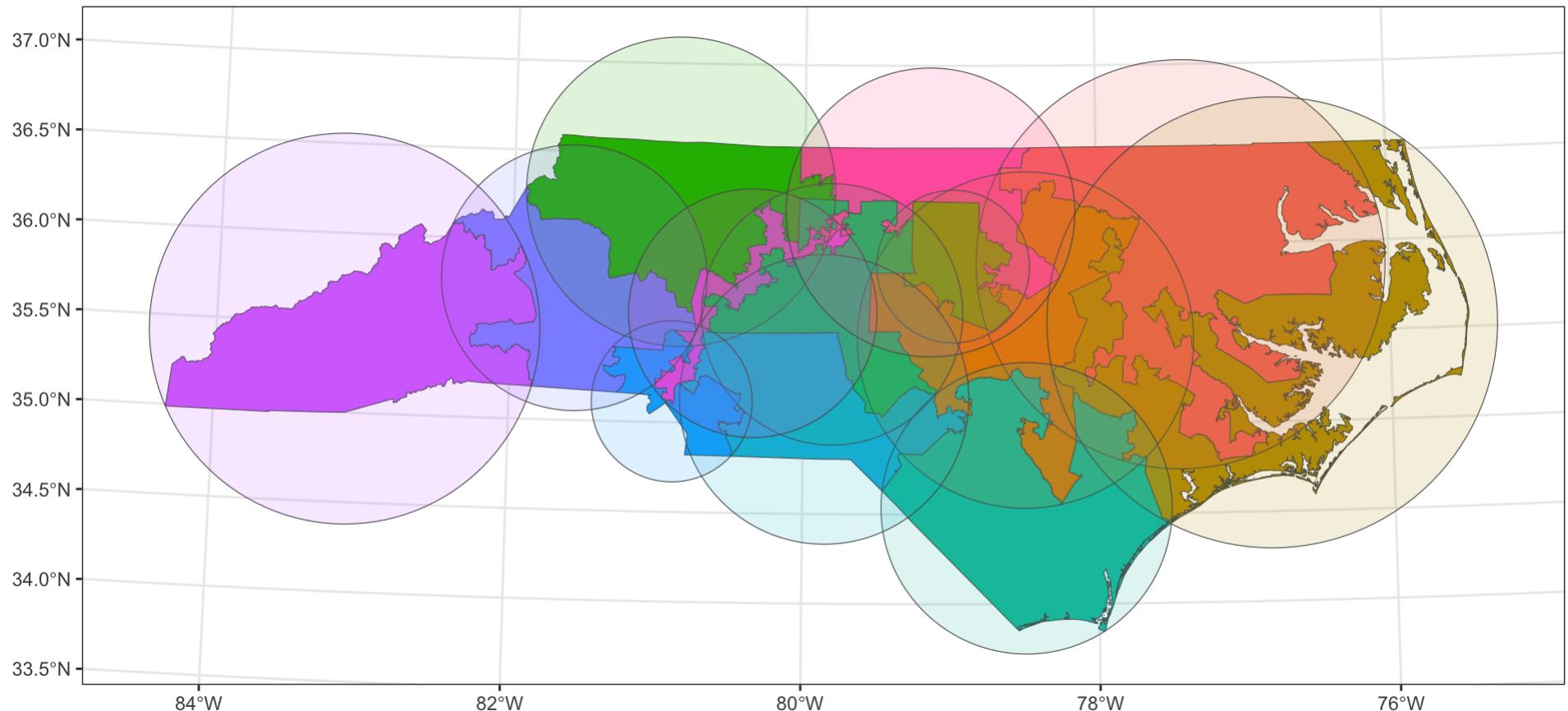
# Measuring Compactness - Reock Score

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

```
1 circs = nc_house |>
2   lwgeom::st_minimum_bounding_circle()
3
4 plot(circs |> filter(DISTRICT == 1) |> st_geometry(), axes=TRUE)
5 plot(nc_house |> select(DISTRICT) |> filter(DISTRICT == 1), add=TRUE)
```

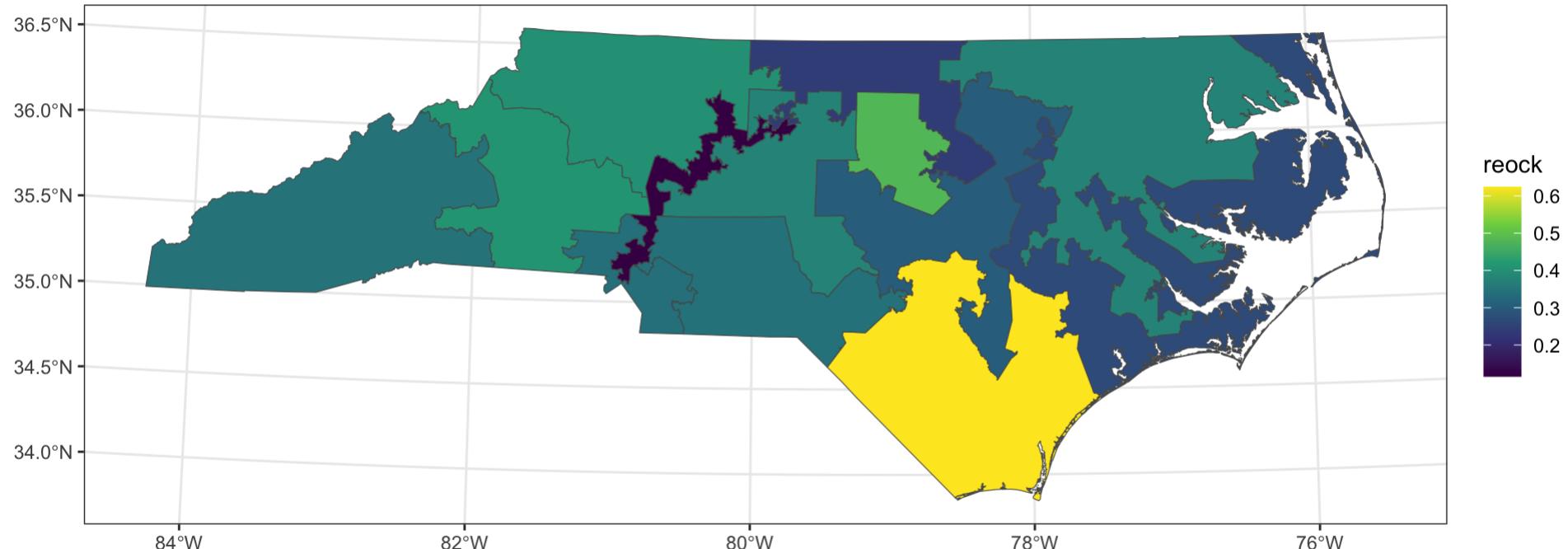


```
1 ggplot(mapping = aes(fill=DISTRICT)) +  
2   geom_sf(data=nc_house) +  
3   geom_sf(data=circs, alpha=0.15) +  
4   guides(color="none", fill="none")
```



# Calculating Reock

```
1 nc_house |>  
2   mutate(reock = (st_area(nc_house) / st_area(circs)) |> as.numeric()) |>  
3   ggplot(aes(fill = reock)) +  
4     geom_sf() +  
5     scale_fill_viridis_c()
```



```

1 nc_house |>
2   mutate(reock = st_area(nc_house) / st_area(circs)) |>
3   arrange(reock) |>
4   print(n=13)

```

Simple feature collection with 13 features and 3 fields

Geometry type: MULTIPOLYGON

Dimension: XY

Bounding box: xmin: 123998.5 ymin: 10979.77 xmax: 930346 ymax: 318095.3

Projected CRS: NAD83(NSRS2007) / North Carolina

# A tibble: 13 × 4

ID	DISTRICT	geom	reock
<chr>	<fct>	<MULTIPOLYGON [m]>	[1]
1	037108112012	((473814.9 211717.3, 472007.4 209951.4, 47...	0.116
2	037108112013	((528146.8 292339.5, 528222.3 292562.4, 52...	0.237
3	037108112003	((911479.9 169543.9, 911621.3 169488.3, 91...	0.266
4	037108112002	((618780.6 152664.8, 618930.2 152932.2, 61...	0.303
5	037108112009	((433786.4 160540.2, 434318.7 158946.9, 43...	0.339
6	037108112008	((615653.2 134126.5, 615013 133993.3, 6147...	0.342
7	037108112011	((154791.1 191470.8, 154769.4 192168.1, 15...	0.344
8	037108112006	((481076.7 216074.5, 481553 216028, 481879...	0.378
9	037108112001	((761514.7 178801.2, 759235.4 176286.8, 76...	0.378
10	037108112005	((529128.7 292213.3, 529538 291136.4, 5294...	0.399
11	037108112010	((424301.6 185435.1, 400728.7 187075, 4018...	0.411
12	037108112004	((629556.3 225844, 629539.1 224734.4, 6281...	0.480
13	037108112007	((739073.9 74030.77, 739510.2 73709.48, 73...	0.624

# Raster Data (stars)

# Example data - Meuse

```
1 ( meuse_rast = stars::read_stars(  
2     system.file("external/test.grd", package="raster")  
3   ) |>  
4     st_transform(st_crs(meuse_riv))  
5   )
```

stars object with 2 dimensions and 1 attribute

attribute(s):

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
test.grd	138.7071	293.9575	371.9001	425.606	501.0102	1736.058	6022

dimension(s):

	from	to	refsys	values	x/y
x	1	80	Amersfoort / RD New [80x115]	178451,...,181611	[x]
y	1	115	Amersfoort / RD New [80x115]	329530,...,334090	[y]

curvilinear grid

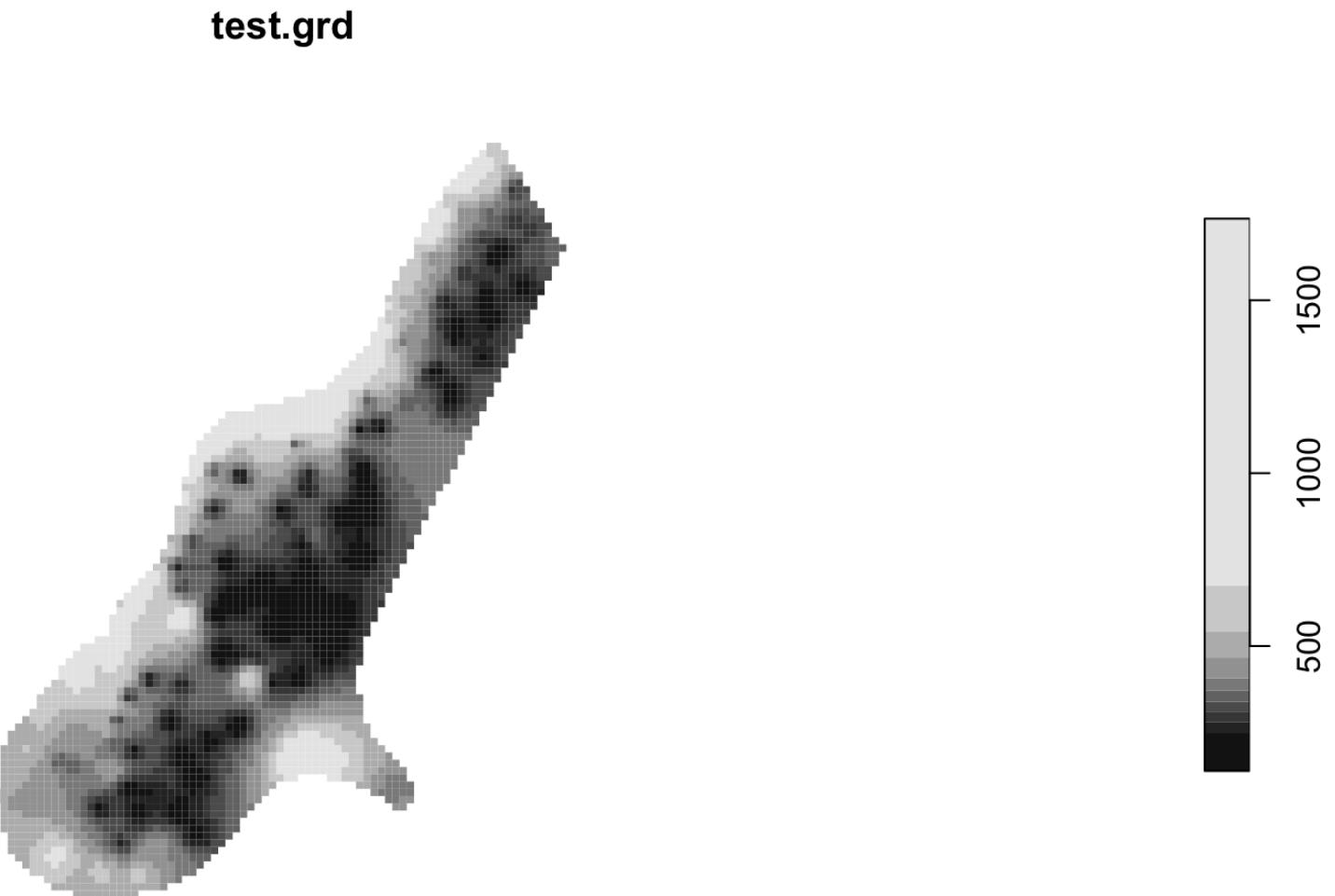
# stars class

```
1 str(meuse_rast)
```

```
List of 1
$ test.grd: num [1:80, 1:115] NA ...
- attr(*, "dimensions")=List of 2
..$ x:List of 7
.. ..$ from : num 1
.. ..$ to : num 80
.. ..$ offset: num NA
.. ..$ delta : num NA
.. ..$ refsys:List of 2
.. .. ..$ input: chr "EPSG:28992"
.. .. ..$ wkt : chr "PROJCRS[\\"Amersfoort / RD New\\",\n      BASEGEOGCRS[\\"Amersfoort\\",\nDATUM[\\"Amersfoort\\",\n          E" | __truncated__
.. .. ...- attr(*, "class")= chr "crs"
.. ..$ point : logi NA
.. ..$ values: num [1:80, 1:115] 178451 178491 178531 178571 178611 ...
.. ..-- attr(*, "class")= chr "dimension"
..$ y:List of 7
.. ..$ from : num 1
.. ..$ to : num 115
.. ..$ offset: num NA
.. ..$ delta : num NA
  ^-----^
```

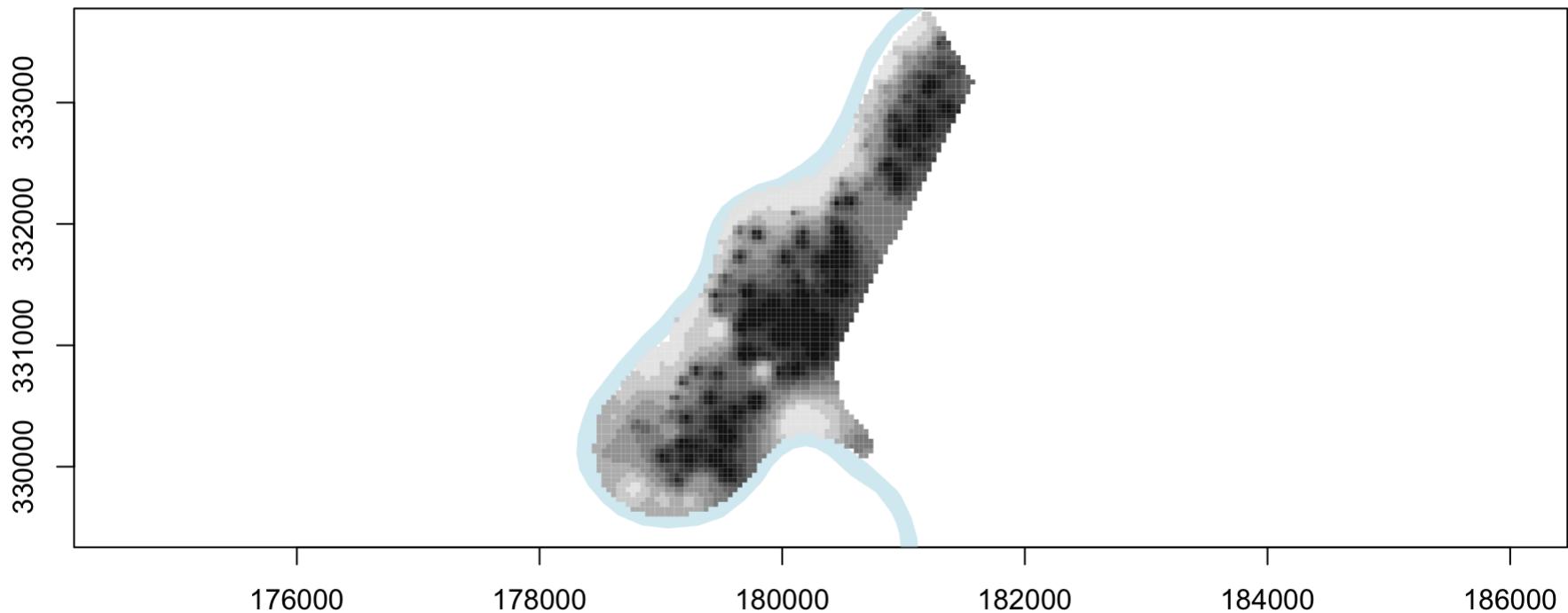
# Plotting

```
1 plot(meuse_rast)
```



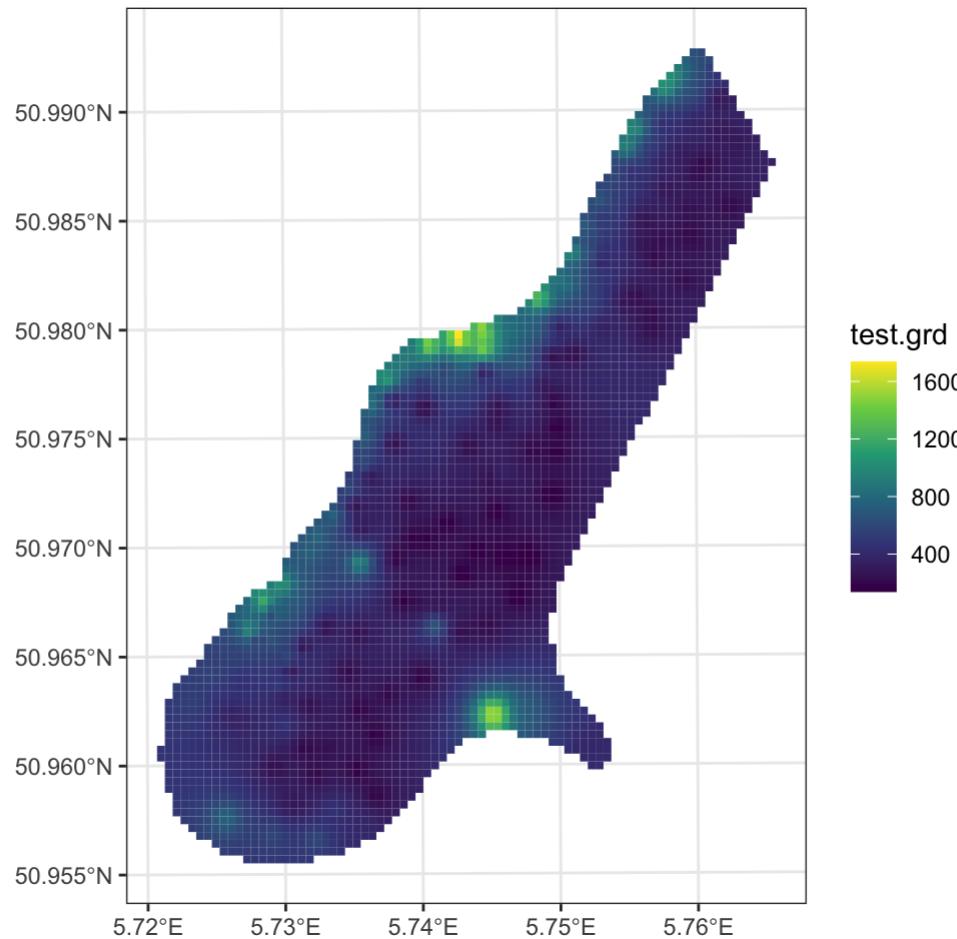
```
1 plot(  
2   meuse_riv,  
3   col=adjustcolor("lightblue",alpha.f = 0.5), border=NA,  
4   ylim = c(329500, 333611), axes=TRUE  
5 )  
6 plot(meuse_rast, add=TRUE)
```

**test.grd**

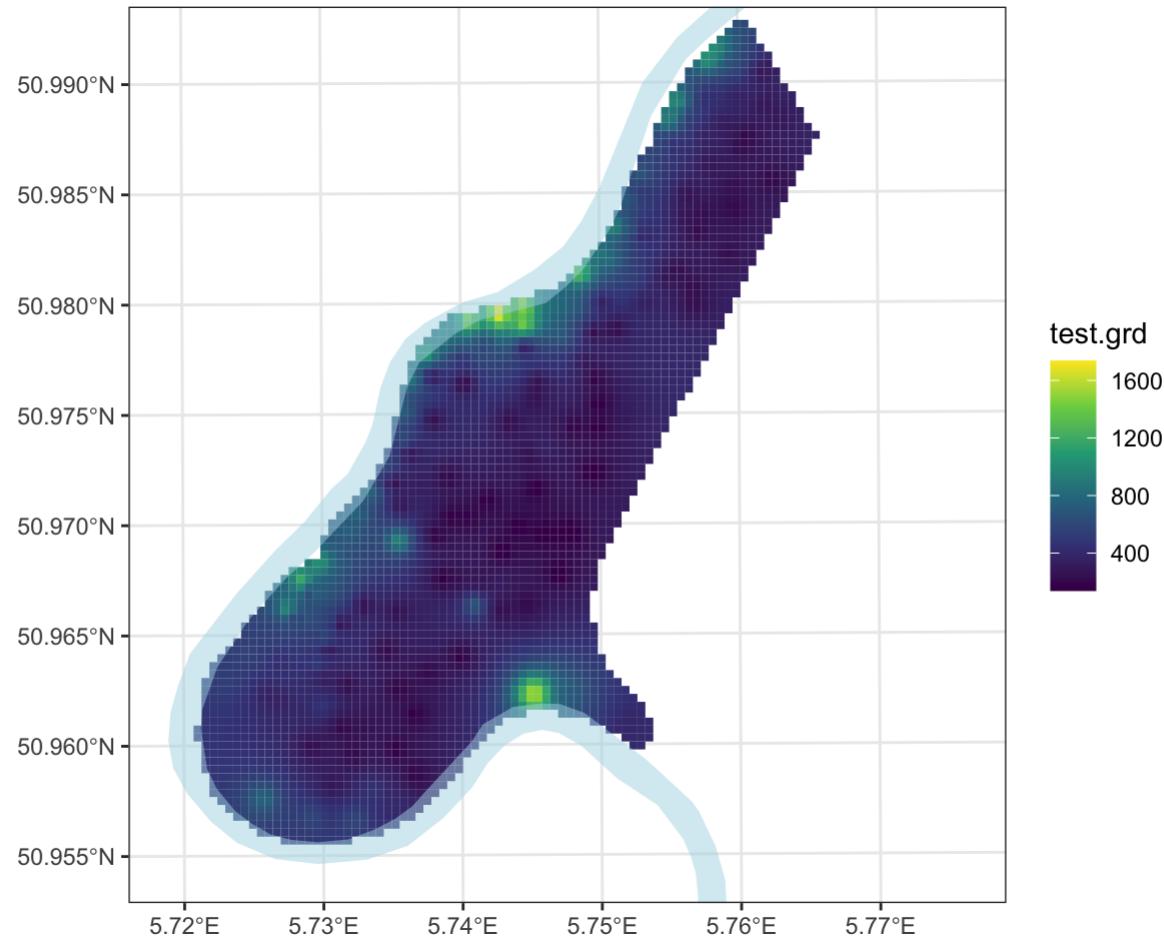


# ggplot

```
1 ggplot() +  
2   stars::geom_stars(data=meuse_rast) +  
3   scale_fill_viridis_c()
```



```
1 ggplot() +  
2   stars::geom_stars(data=meuse_rast) +  
3   geom_sf(data=meuse_riv, fill="lightblue", color=NA, alpha=0.5) +  
4   scale_fill_viridis_c() +  
5   ylim(329500, 333611)
```

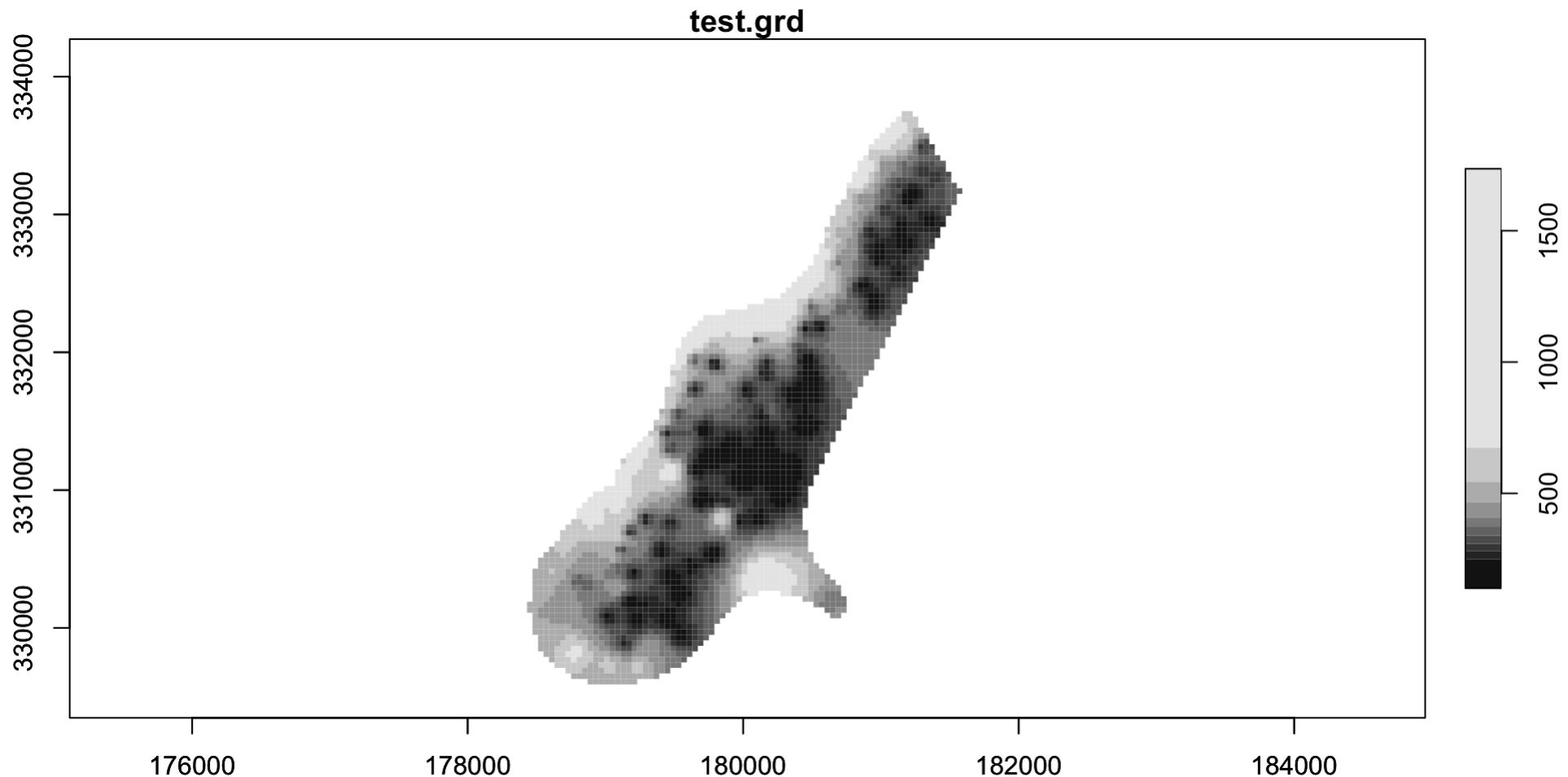


# Rasters and Projections

EPSG 3631

Lat / Long

```
1 plot(meuse_rast, axes=TRUE)
```



```
1 meuse_rast
```

stars object with 2 dimensions and 1 attribute

attribute(s):

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
test.grd	138.7071	293.9575	371.9001	425.606	501.0102	1736.058	6022

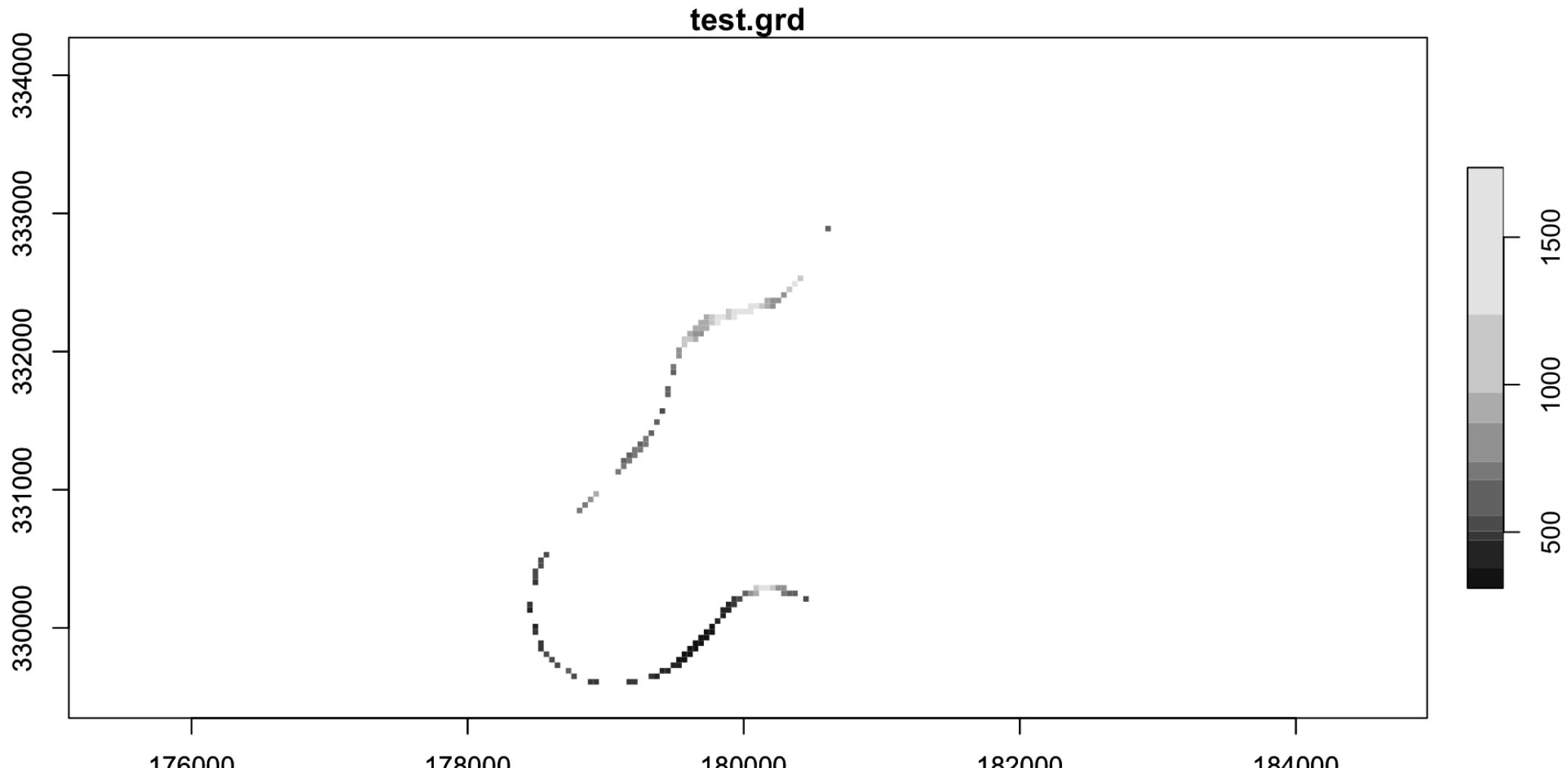
dimension(s):

	from	to	refsys	values	x/y
x	1	80	Amersfoort / RD New [80x115]	178451, ..., 181611	[x]
y	1	115	Amersfoort / RD New [80x115]	329530, ..., 334090	[y]

curvilinear grid

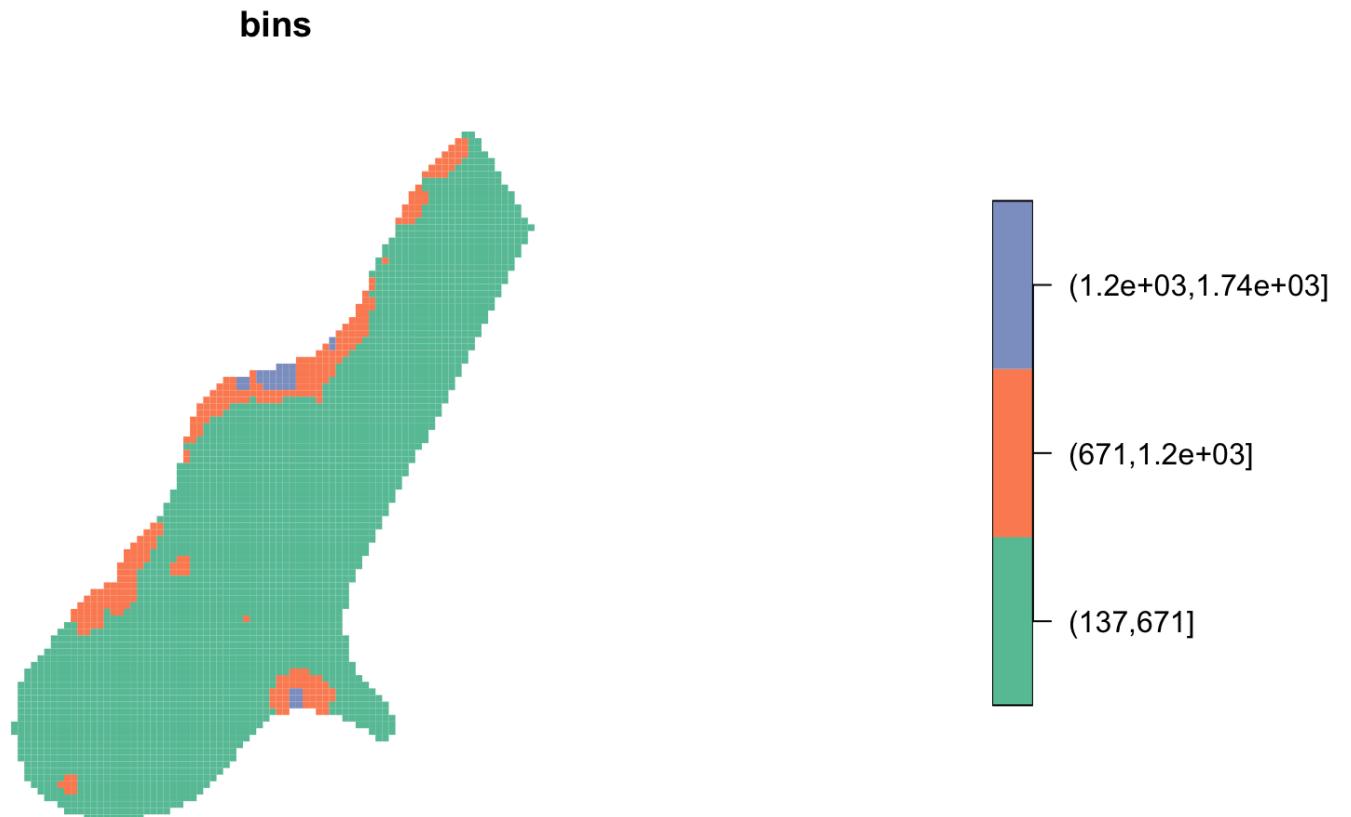
# Cropping

```
1 meuse_rast_riv = meuse_rast[ meuse_riv ]
2 plot(meuse_rast_riv, axes=TRUE)
```



# Segmentation

```
1 meuse_rast |>  
2   mutate(bins = cut(test.grd, 3) ) |>  
3   select(bins) |>  
4   plot()
```

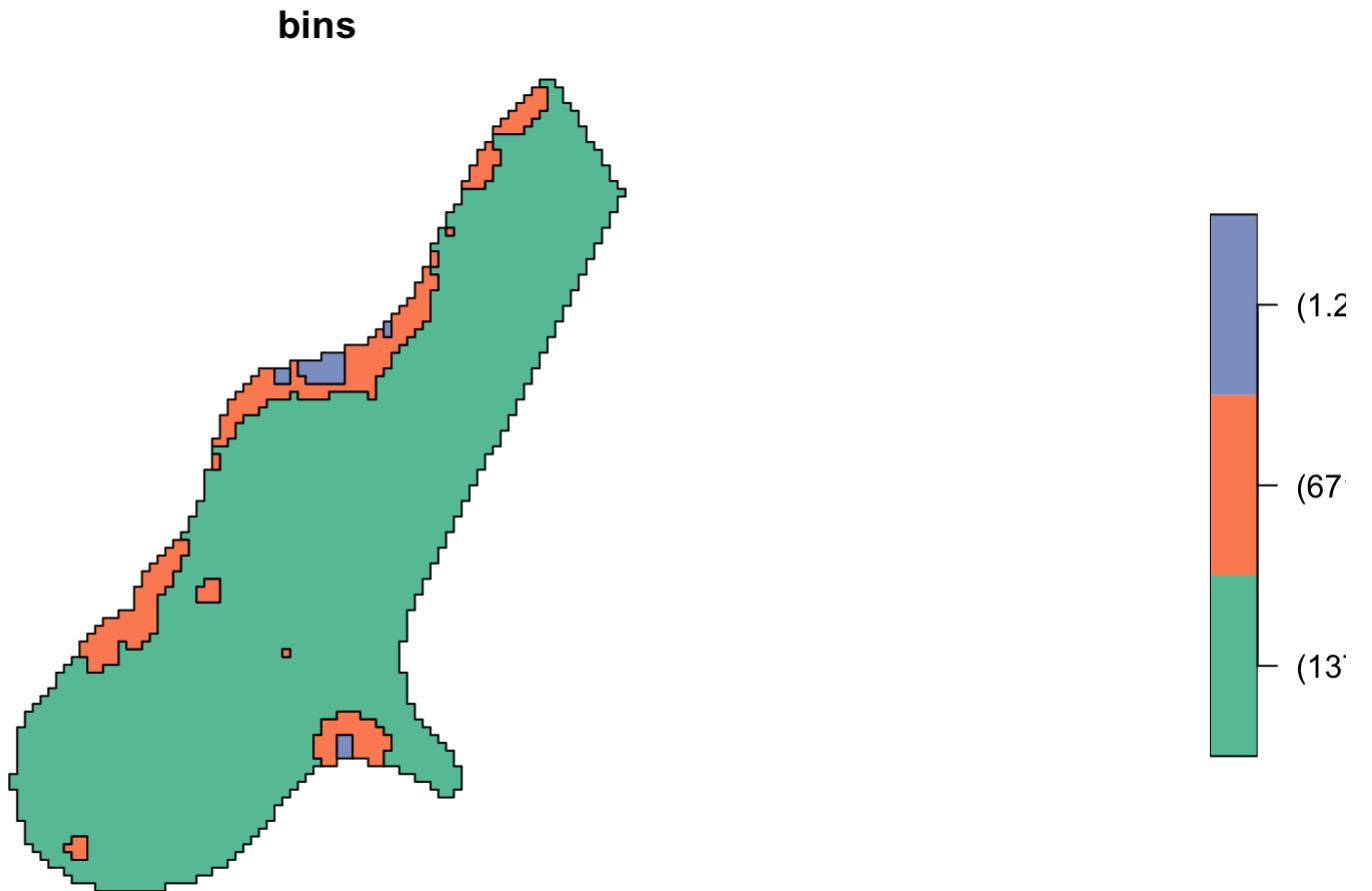


# Polygonization

```
1 meuse_rast_poly = meuse_rast |>
2   mutate(bins = cut(test.grd, 3) ) |>
3   select(bins) |>
4   st_as_sf()
5 plot(meuse_rast_poly)
```



```
1 meuse_rast_poly |>  
2   group_by(bins) |>  
3   summarize() |>  
4   plot()
```



```
1 meuse_rast_poly |>
2   group_by(bins) |>
3   summarize() |>
4   mutate(area = st_area(geometry))
```

Simple feature collection with 3 features and 2 fields

Geometry type: GEOMETRY

Dimension: XY

Bounding box: xmin: 178431 ymin: 329589.8 xmax: 181590.9 ymax: 333749.8

Projected CRS: Amersfoort / RD New

# A tibble: 3 × 3

bins		geometry	area
*	<fct>	<GEOMETRY [m]>	[m^2]
1	(137,671]	POLYGON ((178551 329829.8, 178511 329829.8, 17...	4.56e6
2	(671,1.2e+03]	MULTIPOLYGON (((178711 329829.8, 178751 329829...	4.74e5
3	(1.2e+03,1.74e+03]	MULTIPOLYGON (((179790.9 332189.8, 179790.9 33...	5.12e4

