

# Spatial Data and Cartography (Part 2)

Lecture 17

Dr. Colin Rundel

# Plotting

# Example Data - NC SIDS

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

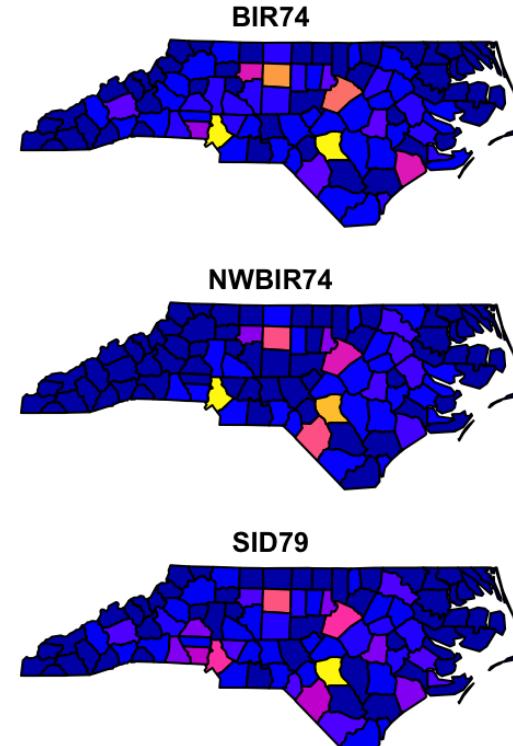
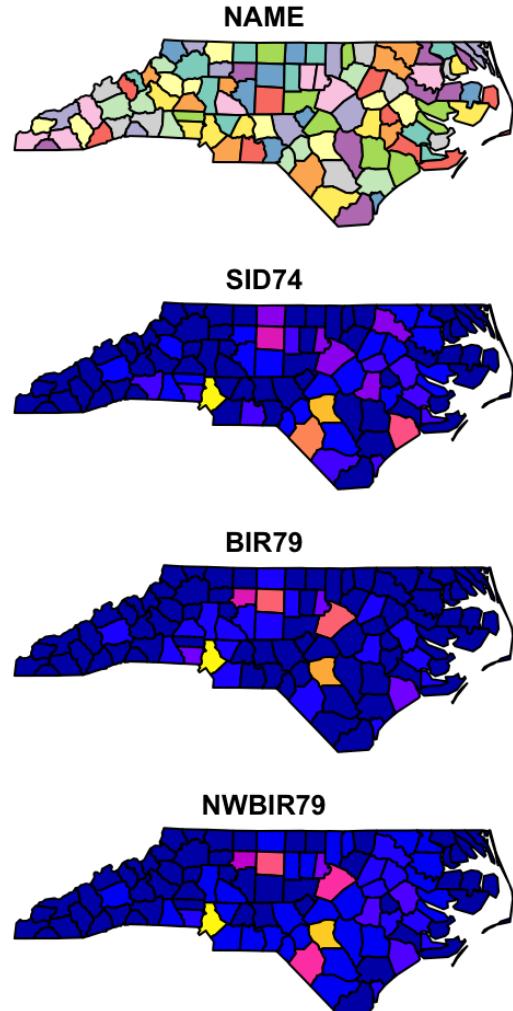
Geodetic CRS: NAD27

# 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

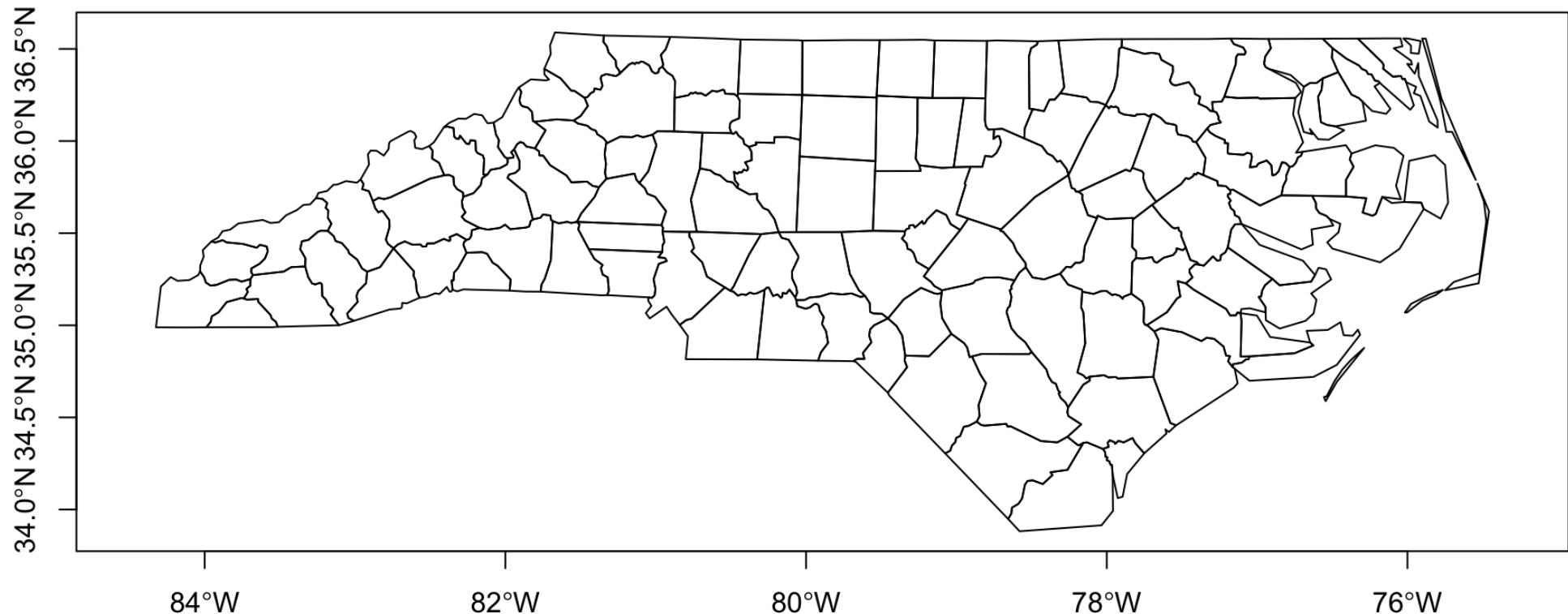
# Base Plots

```
1 plot(nc)
```



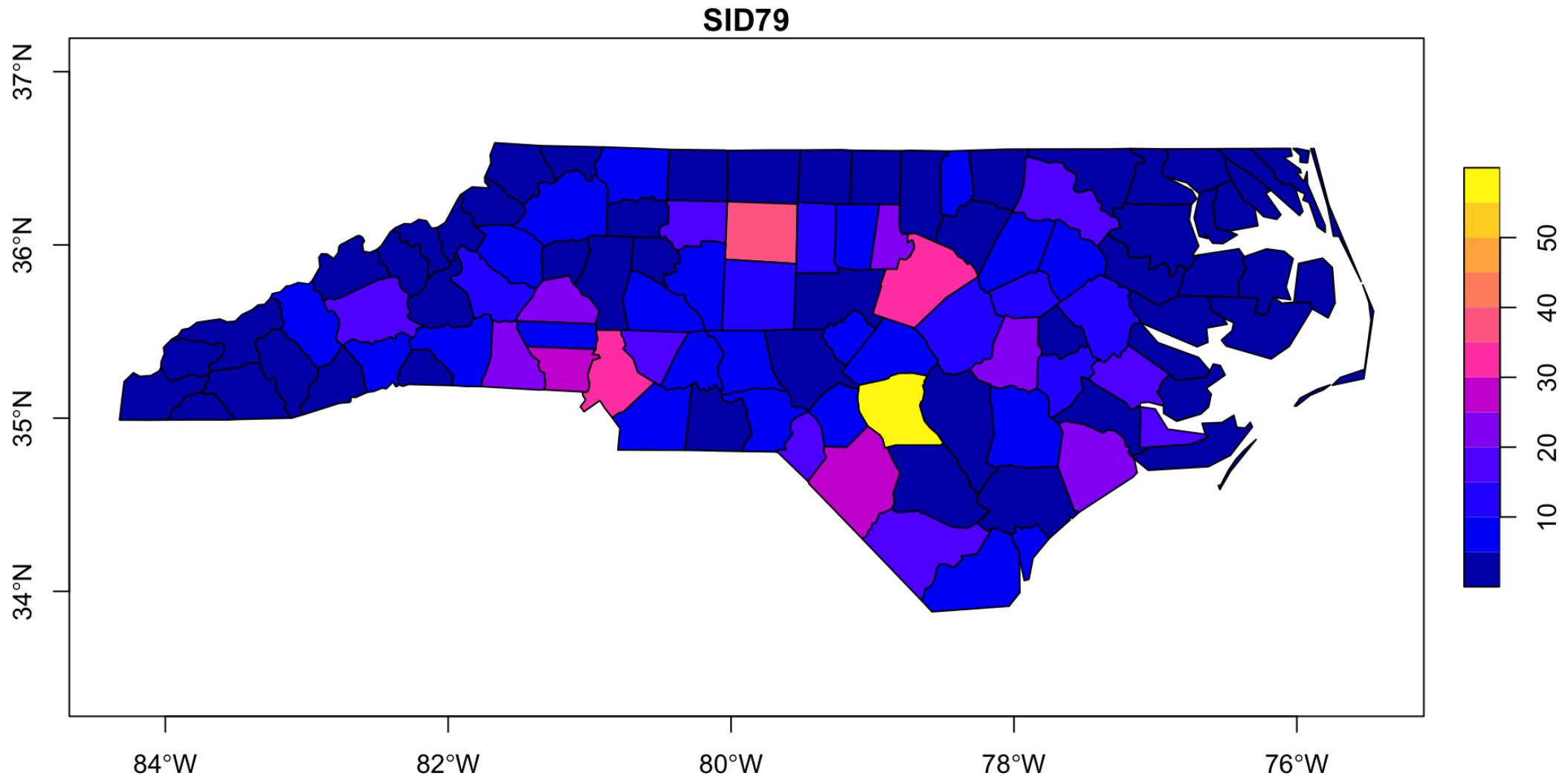
# Geometry Plot

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



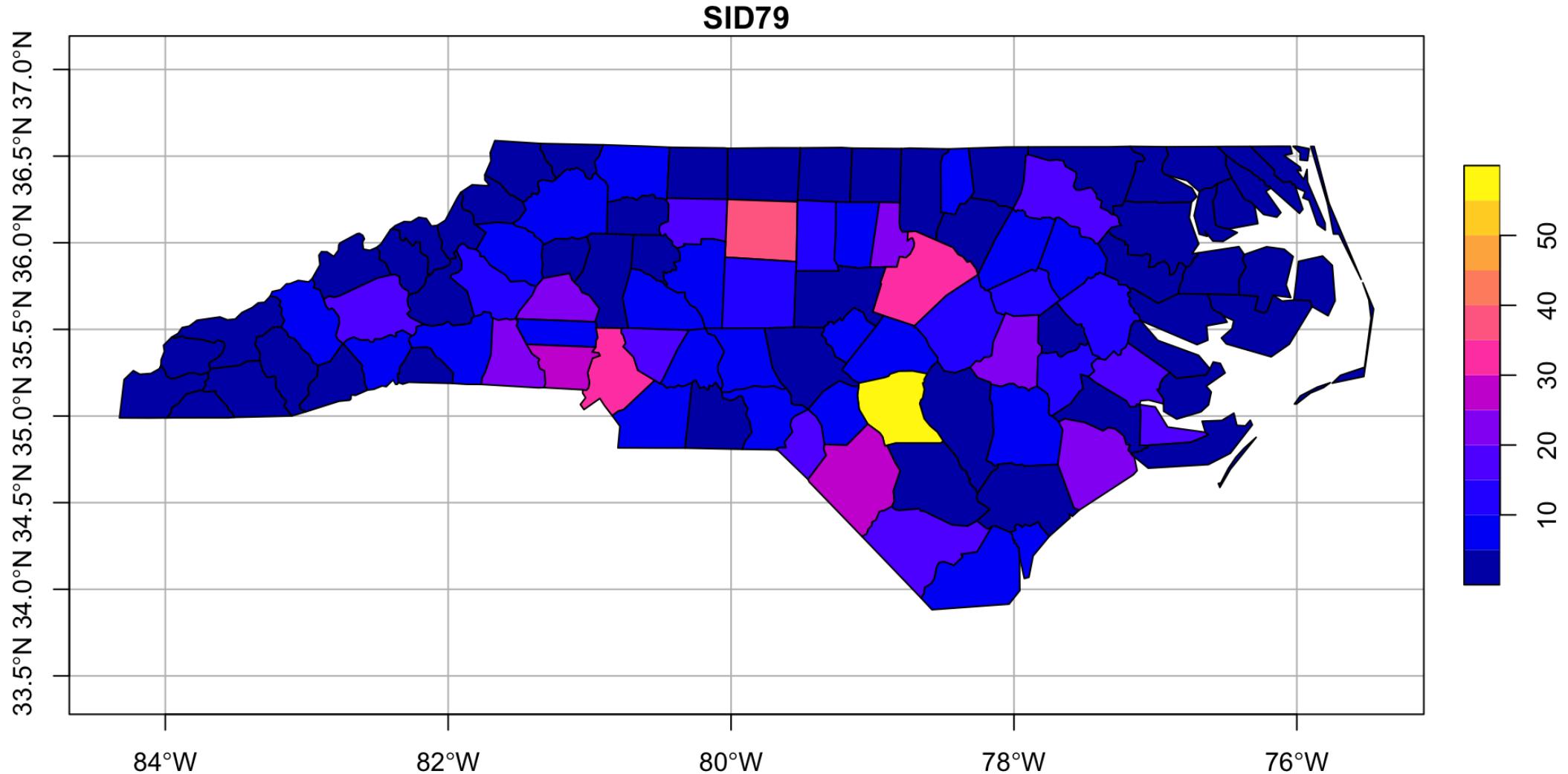
# Graticules

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



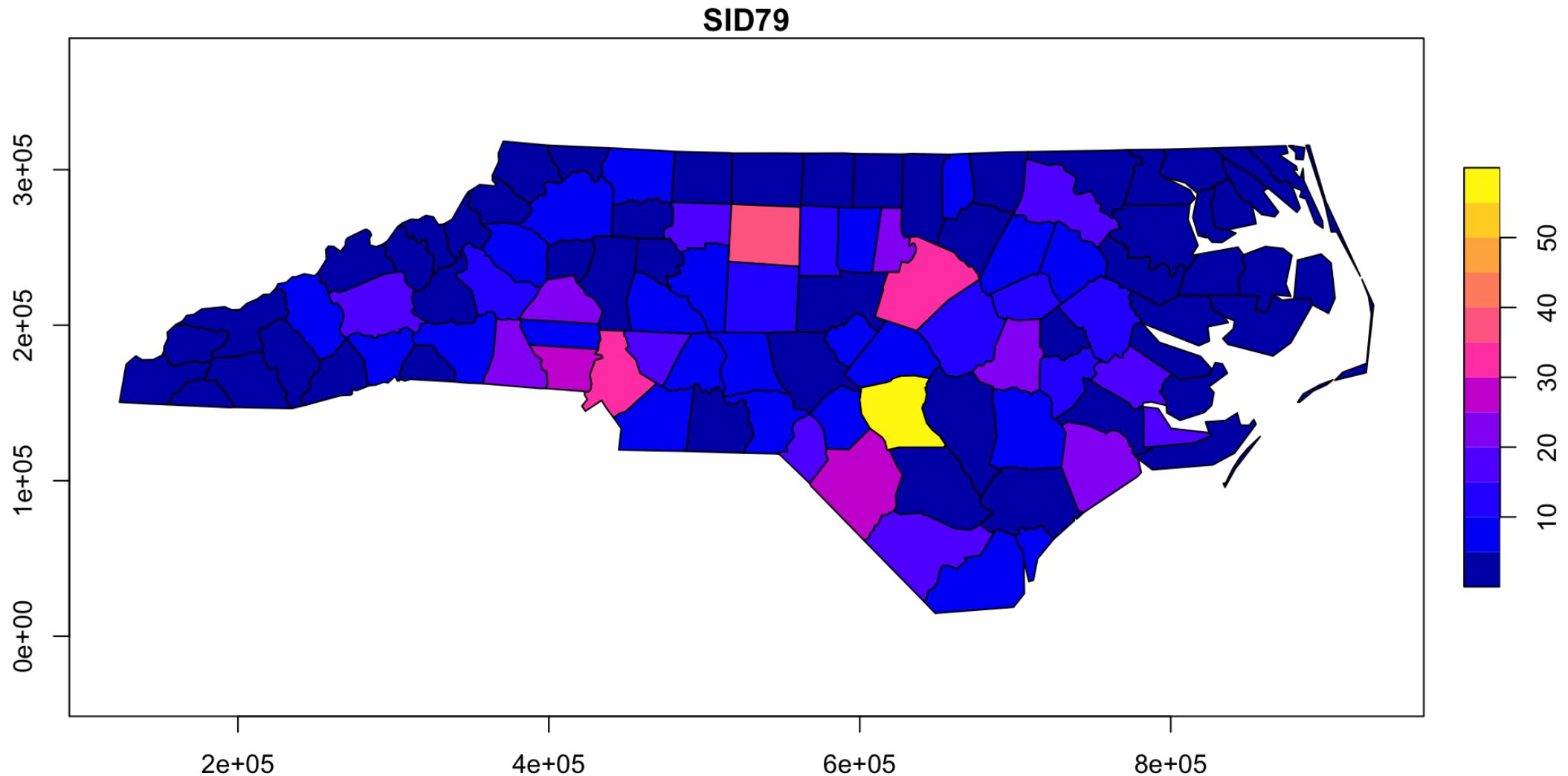
# Graticules

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



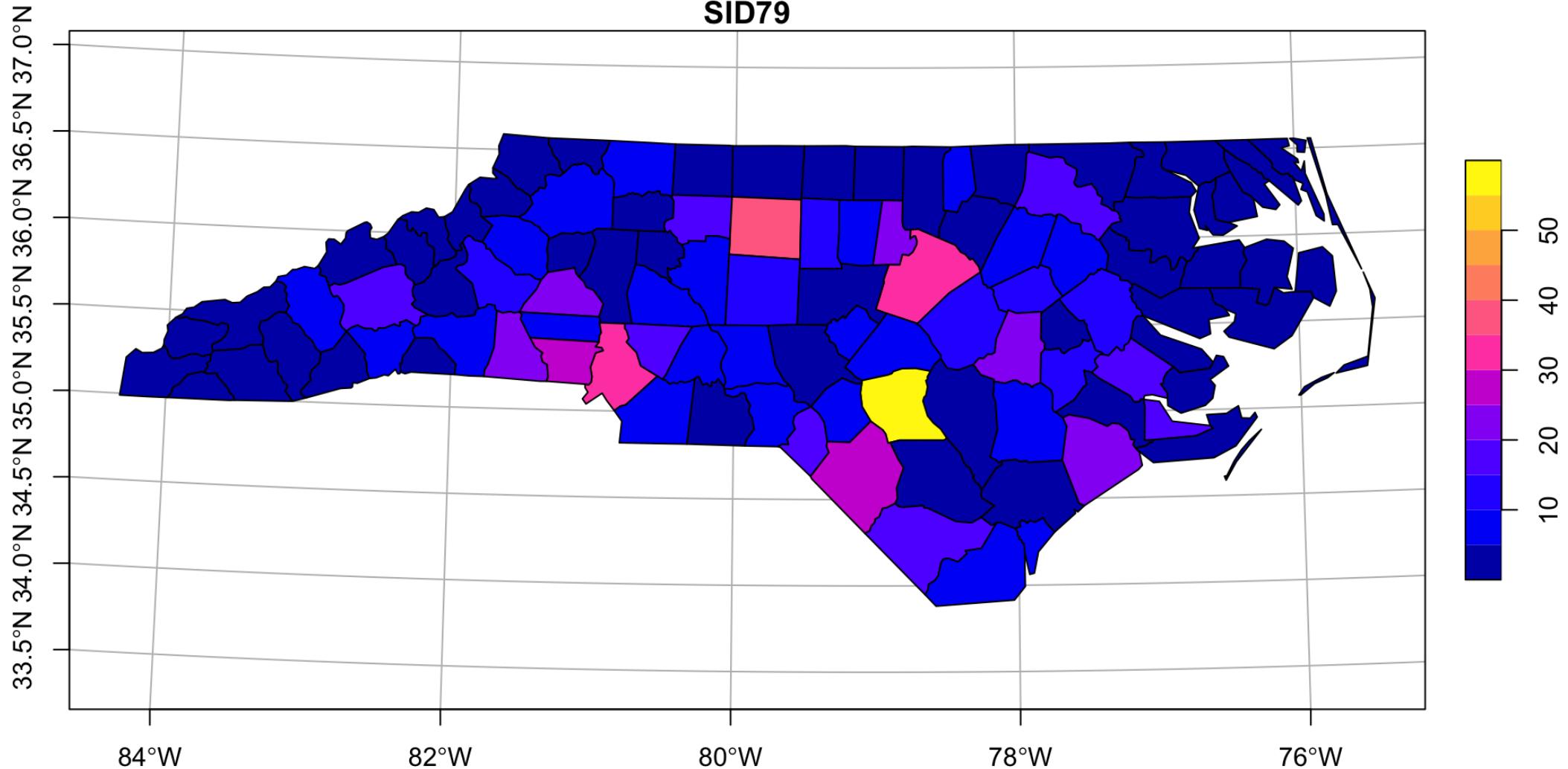
# Graticules (EPSG:3631)

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



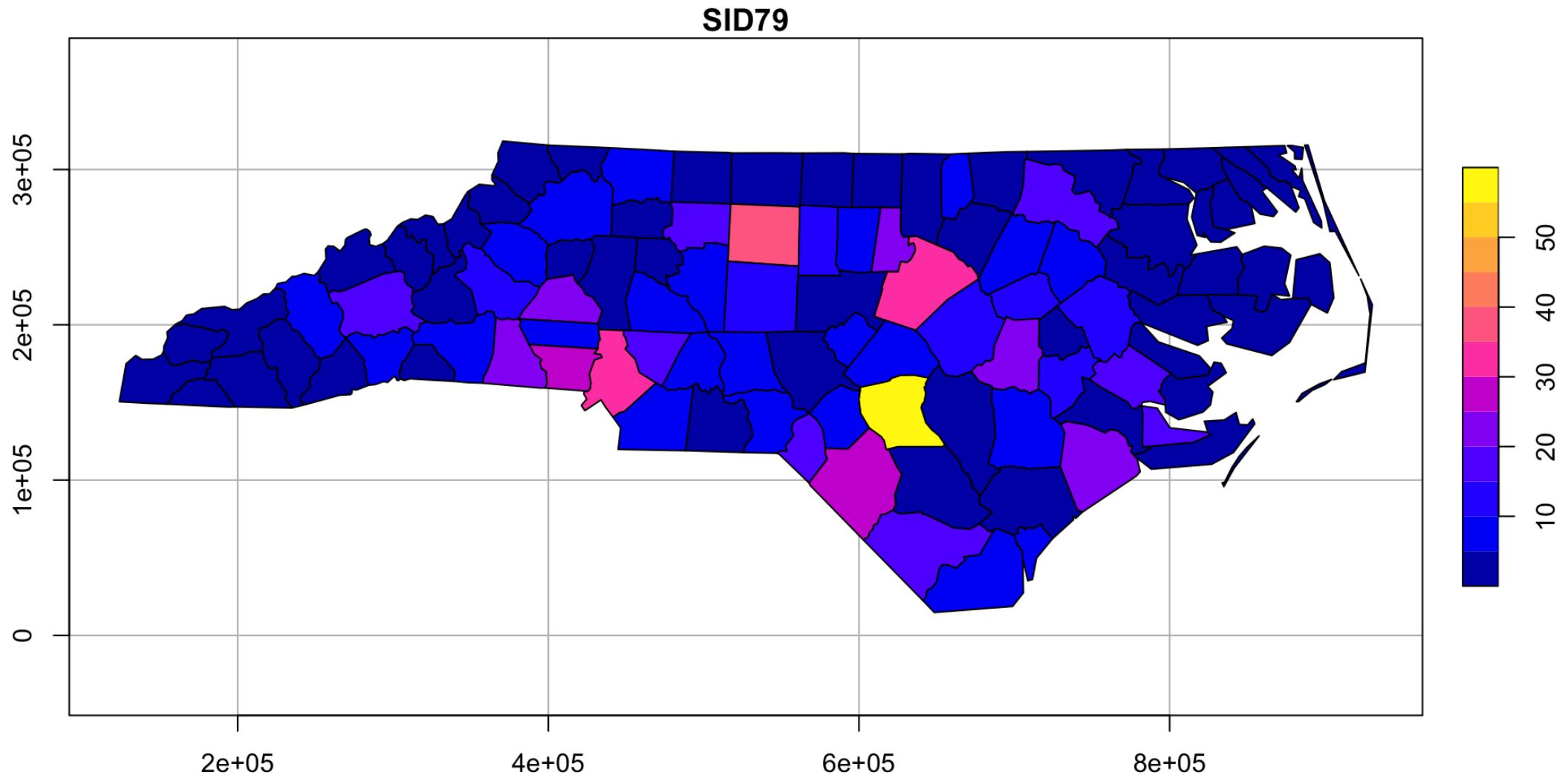
# Graticules (EPSG:3631)

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



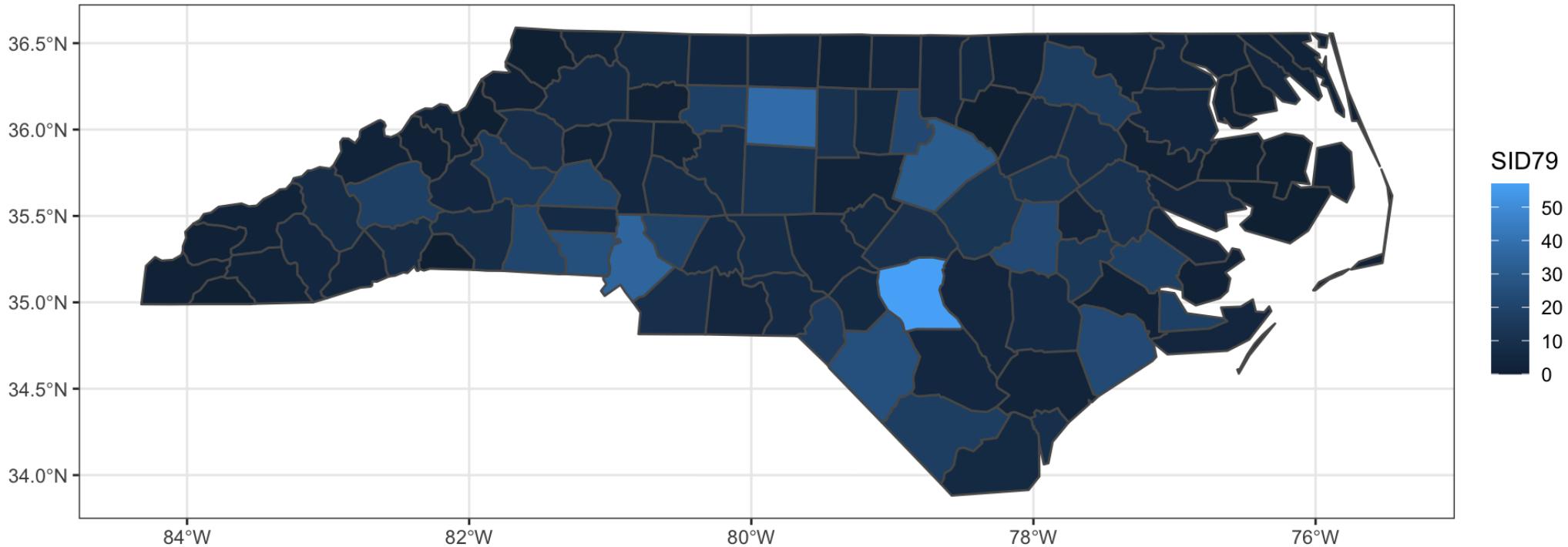
# Graticules (EPSG:3631)

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



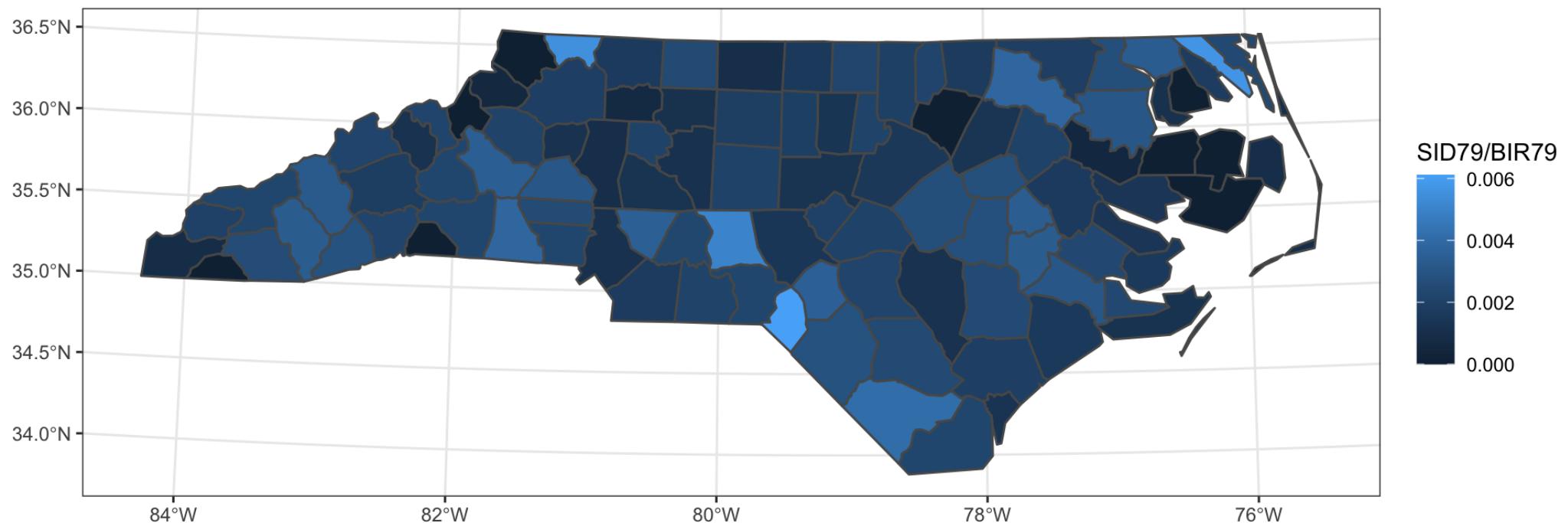
# ggplot2

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



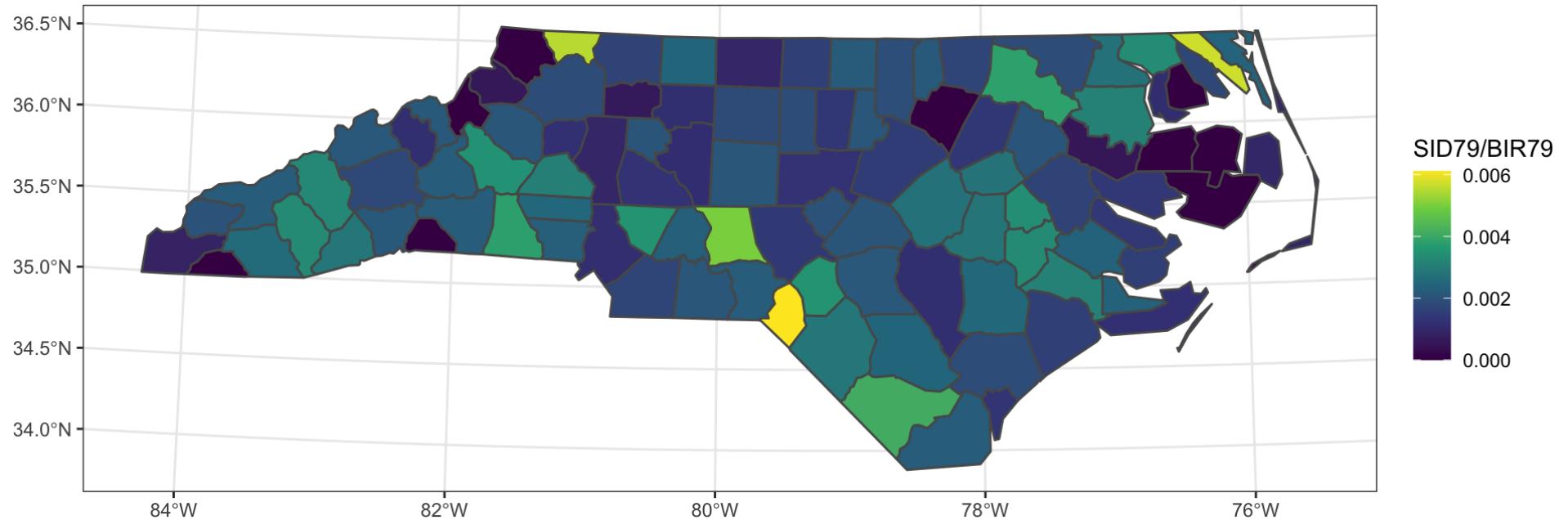
# ggplot2 + projections

```
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

```
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

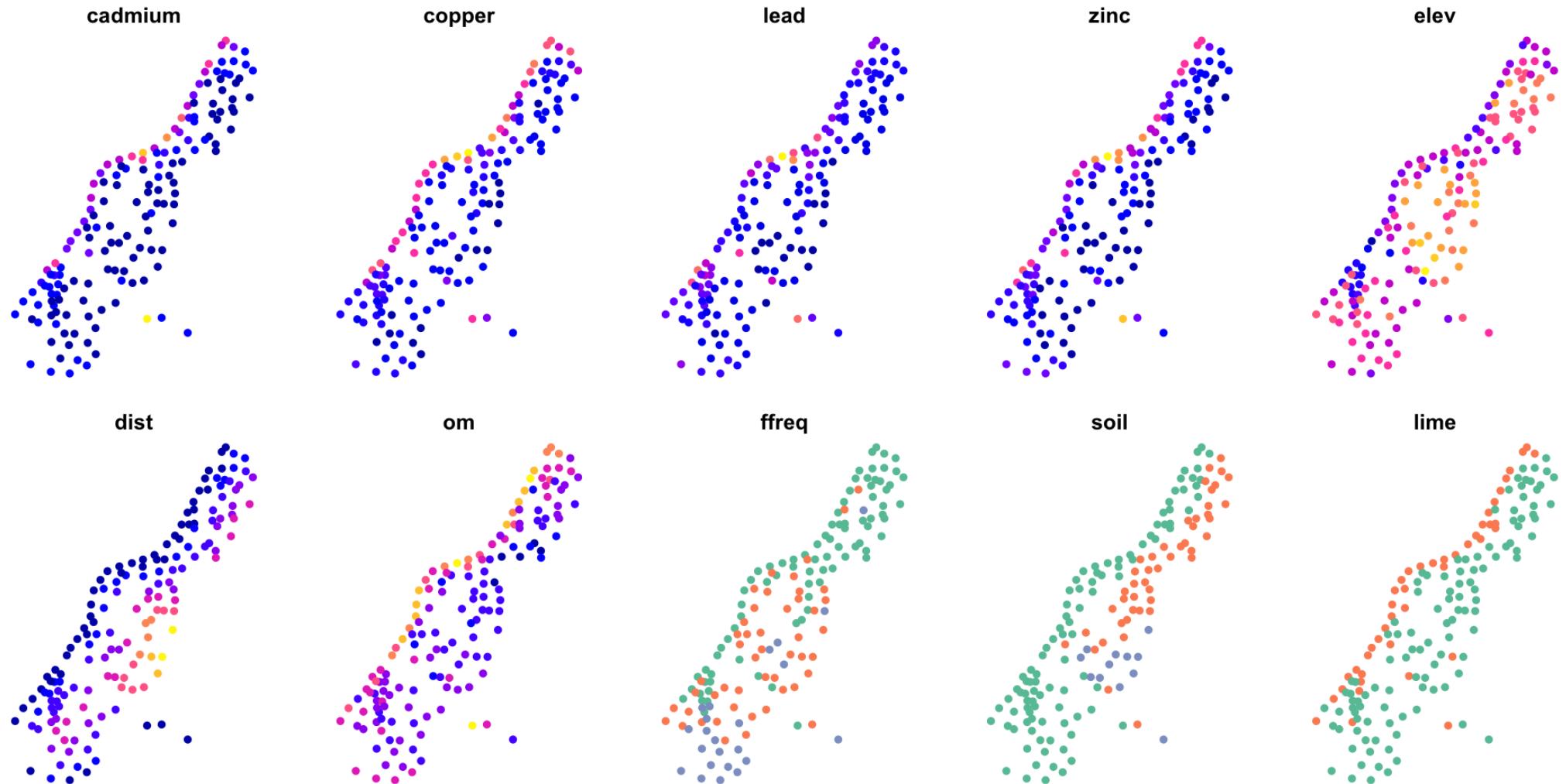
Projected CRS: Amersfoort / RD New

x

1 POLYGON ((182003.7 337678.6...

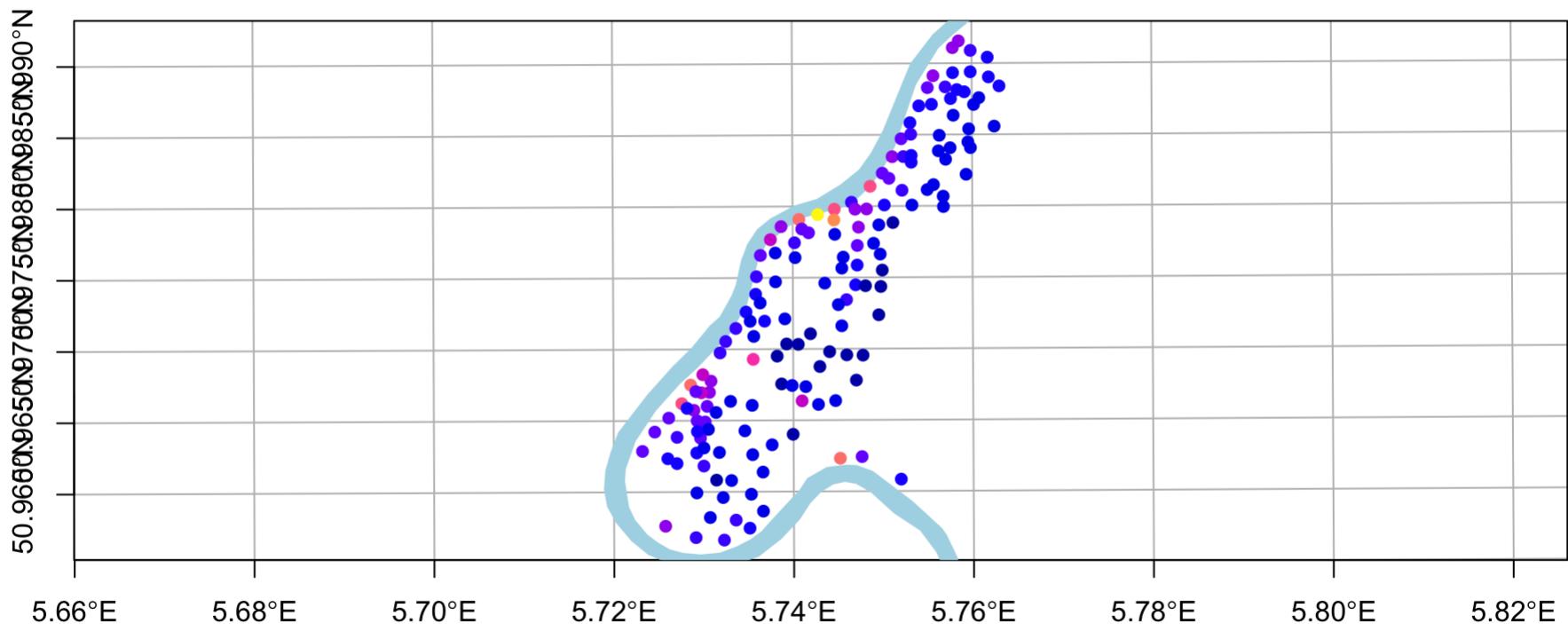
# Meuse

```
1 plot(meuse, pch=16)
```



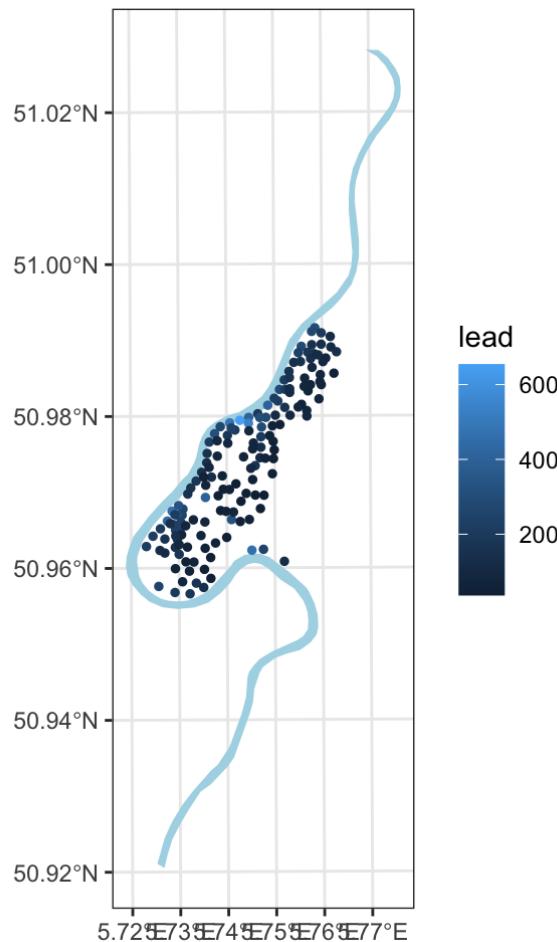
# Layering plots

```
1 plot(meuse_riv, col=adjustcolor("lightblue", alpha.f=1), border = NA,  
2      axes=TRUE, graticule=st_crs(4326),  
3      xlim=st_bbox(meuse)[c(1,3)], ylim=st_bbox(meuse)[c(2,4)])  
4 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)
```



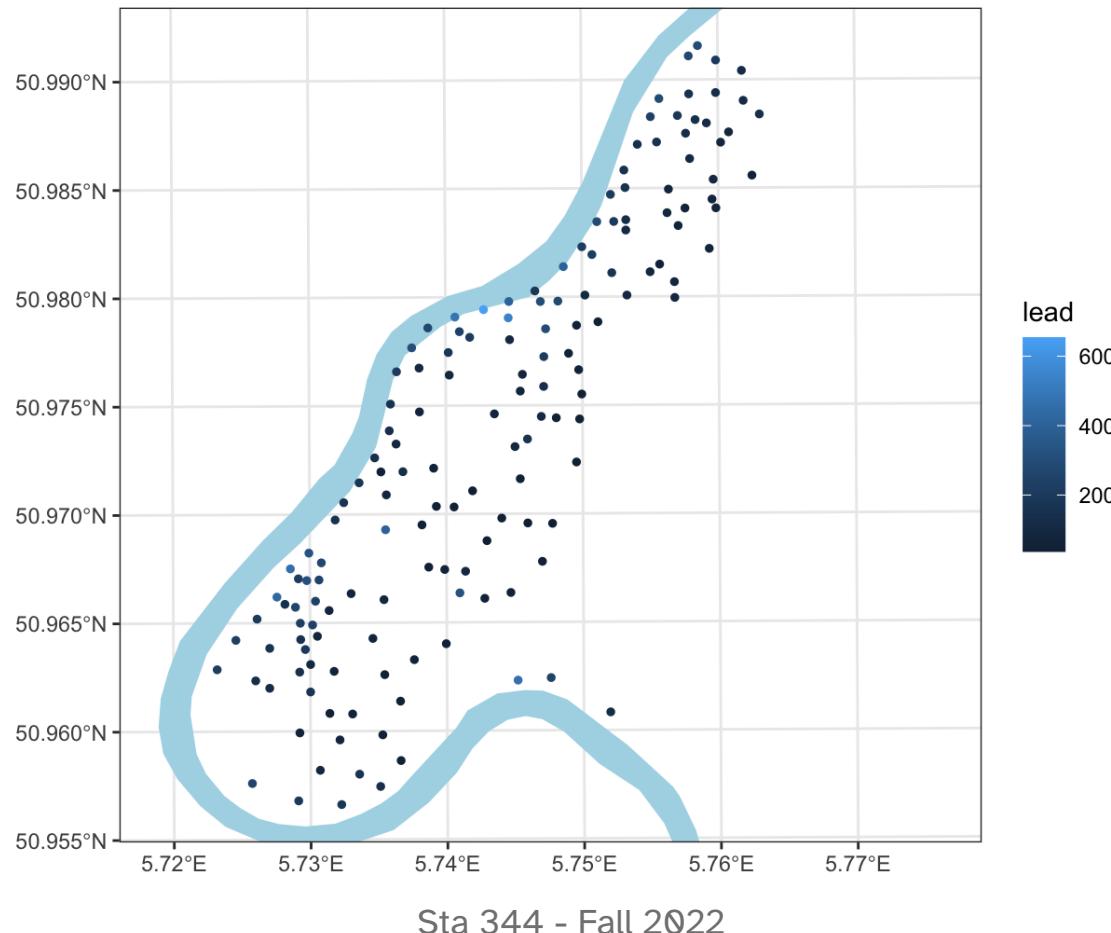
# ggplot2 - axis limits

```
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)
```



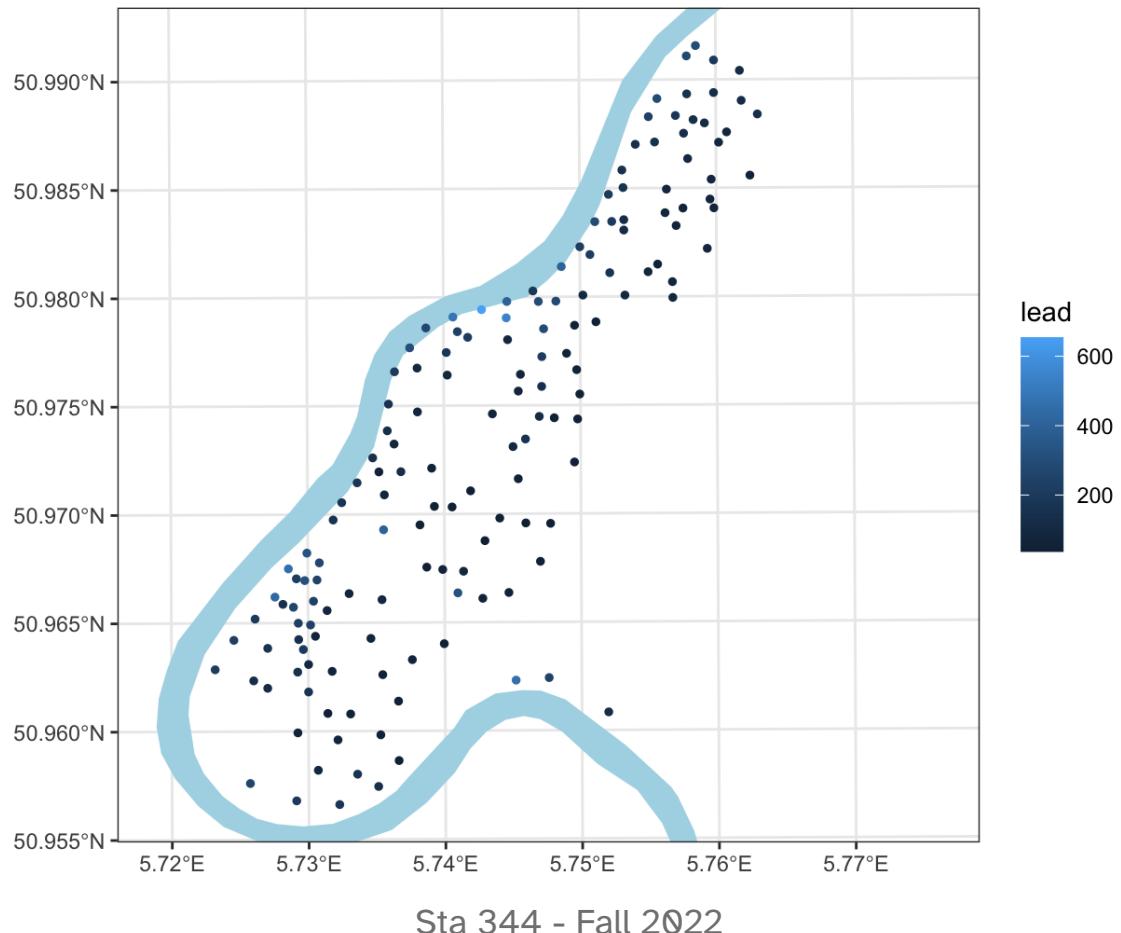
# ggplot2 - axis limits

```
1 ggplot() +  
2   geom_sf(data=meuse_riv, fill="lightblue", color=NA) +  
3   geom_sf(data=meuse, aes(color=lead), size=1) +  
4   ylim(329714, 333611)
```



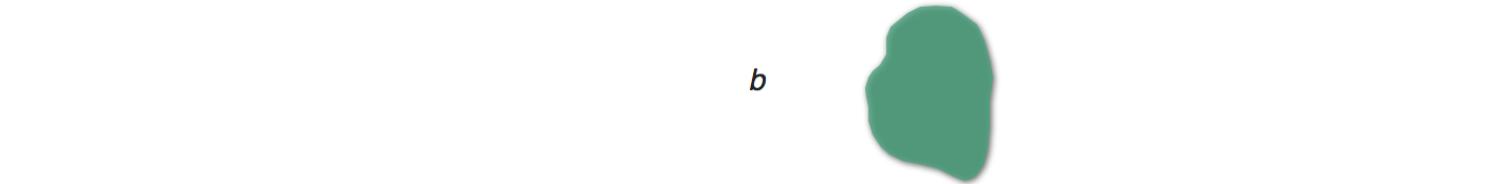
# ggplot2 - bounding box

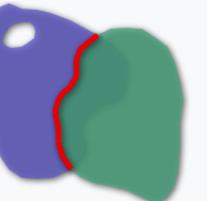
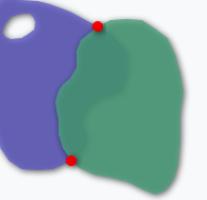
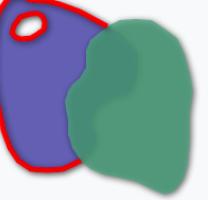
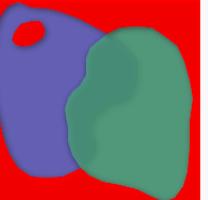
```
1 ggplot() +  
2   geom_sf(data=st_sf(meuse_riv), fill="lightblue", color=NA) +  
3   geom_sf(data=meuse, aes(color=lead), size=1) +  
4   ylim(st_bbox(meuse)[ "ymin" ], st_bbox(meuse)[ "ymax" ])
```



# Geometry Predicates

# DE-9IM



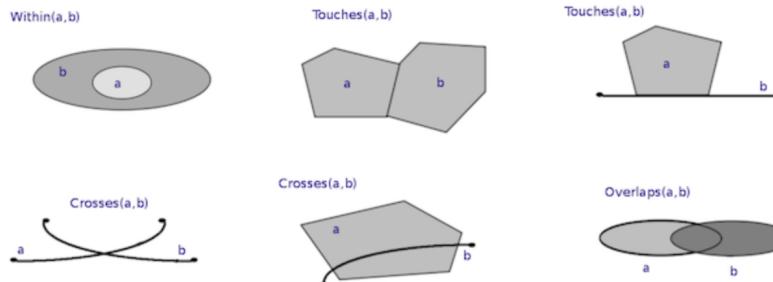
	<b>Interior</b>	<b>Boundary</b>	<b>Exterior</b>
<b>Interior</b>			
<b>Boundary</b>			
<b>Exterior</b>			

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



st\_within(a,b):

$$\begin{bmatrix} T & * & F \\ * & * & F \\ * & * & * \end{bmatrix}$$

st\_touches(a,b):

$$\begin{bmatrix} F & T & * \\ * & * & * \\ * & * & * \end{bmatrix} \cup \begin{bmatrix} F & * & * \\ T & * & * \\ * & * & * \end{bmatrix} \cup \begin{bmatrix} F & * & * \\ * & T & * \\ * & * & * \end{bmatrix}$$

# 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
```

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

# Which counties have an airport?

```
1 ncc |>
2   mutate(
3     airports = st_intersects(ncc, air) |> strip_attrs(),
4     n_airports = purrr::map_int(airports, length)
5   ) |>
6   filter(n_airports > 0)
```

Simple feature collection with 16 features and 10 fields

Geometry type: MULTIPOLYGON

Dimension: XY

Bounding box: xmin: -82.88783 ymin: 33.92835 xmax: -75.46003 ymax: 36.26165

Geodetic CRS: NAD83

# A tibble: 16 × 11

	AREA	PERIM... <sup>1</sup>	COUNT... <sup>2</sup>	STATE	COUNTY	FIPS	STATE... <sup>3</sup>	SQUAR... <sup>4</sup>
*	<dbl>	<dbl>	<dbl>	<chr>	<chr>	<chr>	<chr>	<dbl>
1	0.107	1.54	2065	NC	Forsy...	37067	37	413.
2	0.170	1.69	2069	NC	Guilf...	37081	37	658.
3	0.0996	6.15	2077	NC	Dare ...	37055	37	386.
4	0.221	2.14	2106	NC	Wake ...	37183	37	858.
5	0.169	2.12	2151	NC	Pitt ...	37147	37	654.
6	0.106	1.56	2152	NC	Cataw...	37035	37	413.
7	0.170	2.37	2153	NC	Bunco...	37021	37	660.
8	0.143	1.71	2197	NC	Wavne...	37191	37	557.

# Which counties are adjacent to Durham County?

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

Simple feature collection with 5 features and 10 fields

Geometry type: MULTIPOLYGON

Dimension: XY

Bounding box: xmin: -79.55573 ymin: 35.51505 xmax: -78.25503 ymax: 36.54245

Geodetic CRS: NAD83

# A tibble: 5 × 11

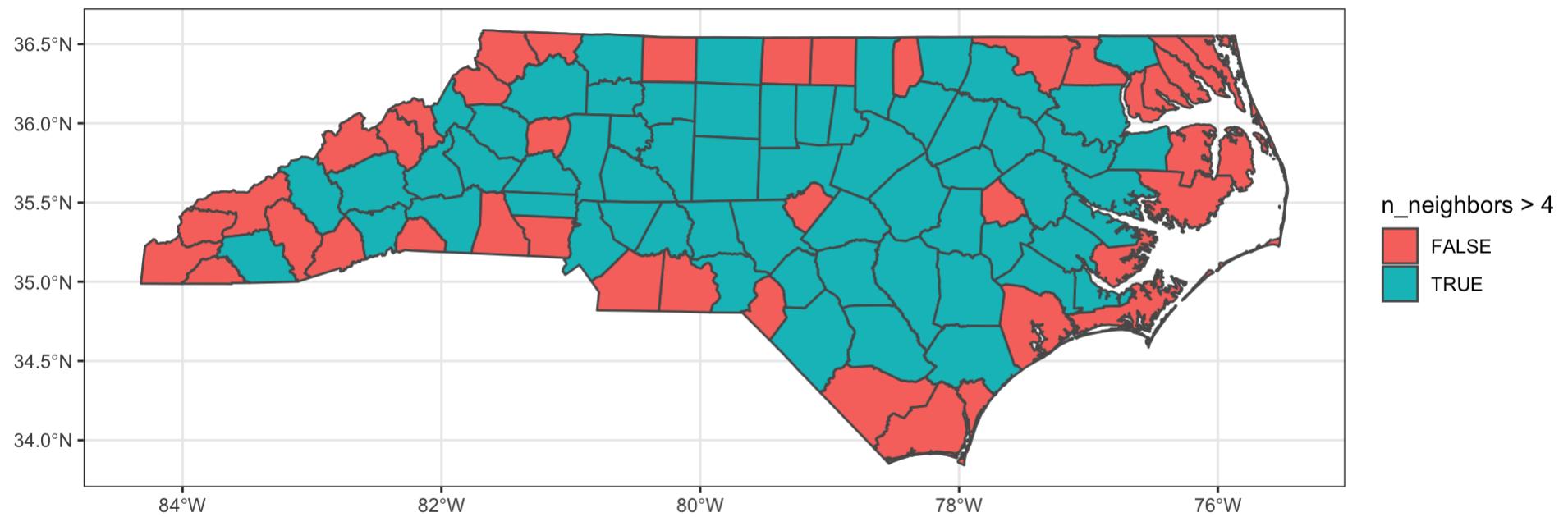
	AREA	PERIMETER	COUNTY	STATE	COUNTY	FIPS	STATE	SQUAREMILE
*	<dbl>	<dbl>	<dbl>	<chr>	<chr>	<chr>	<chr>	<dbl>

```
1 0.105      1.30    2006 NC Perso... 37145 37        404.
2 0.139      1.69    2008 NC Granv... 37077 37        536.
3 0.104      1.30    2074 NC Orang... 37135 37        401.
4 0.221      2.14    2106 NC Wake ... 37183 37        858.
5 0.183      2.26    2141 NC Chath... 37037 37        710.
```

# ... with 3 more variables: geometry <MULTIPOLYGON [°]>,  
# touch\_durham <list>, n\_touches <int>, and abbreviated  
# variable names ¹COUNTYPO10. ²STATE FIPS ³SQUAREMILE

# Which counties have more than 4 neighbors?

```
1 ncc |>
2   mutate(
3     neighbors = st_touches(ncc) |> strip_attrs(),
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

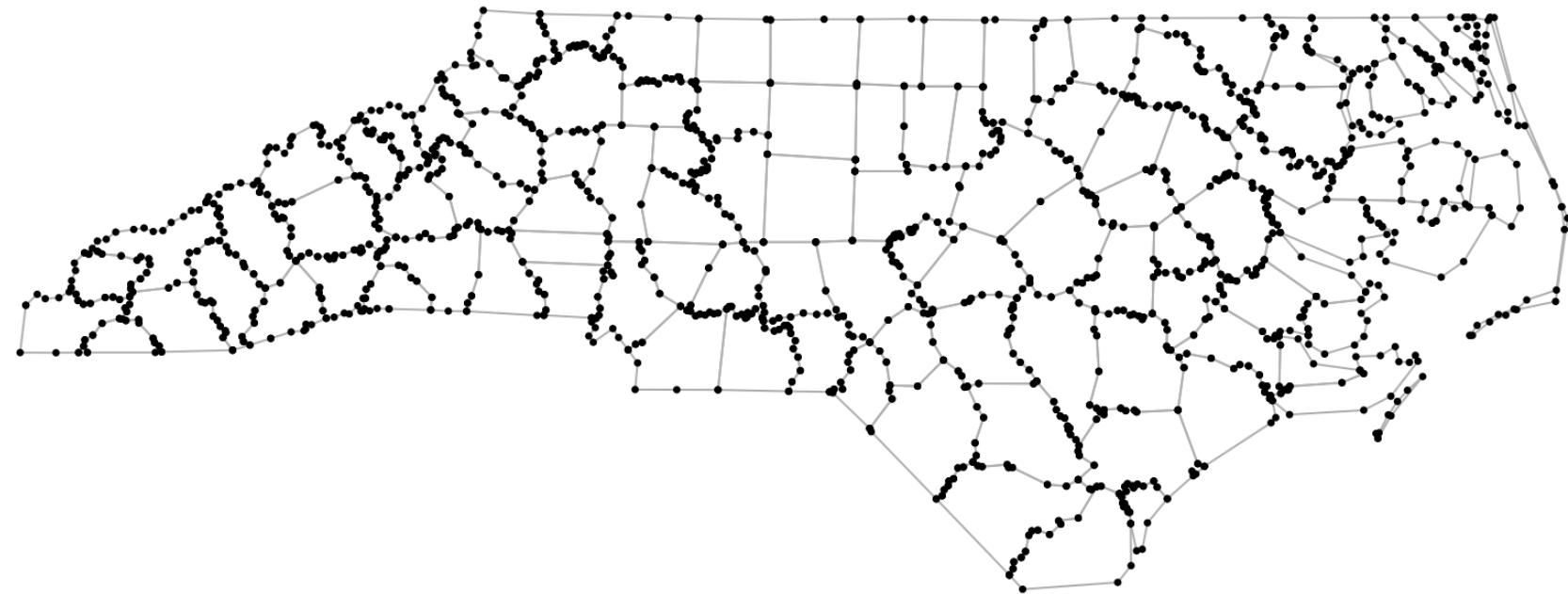
Bounding box: xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax:  
36.58965

Geodetic CRS: NAD27

# A tibble: 100 × 8

	NAME	BIR74	SID74	NWBIR74	BIR79	SID79	NWBIR79
	<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

```
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

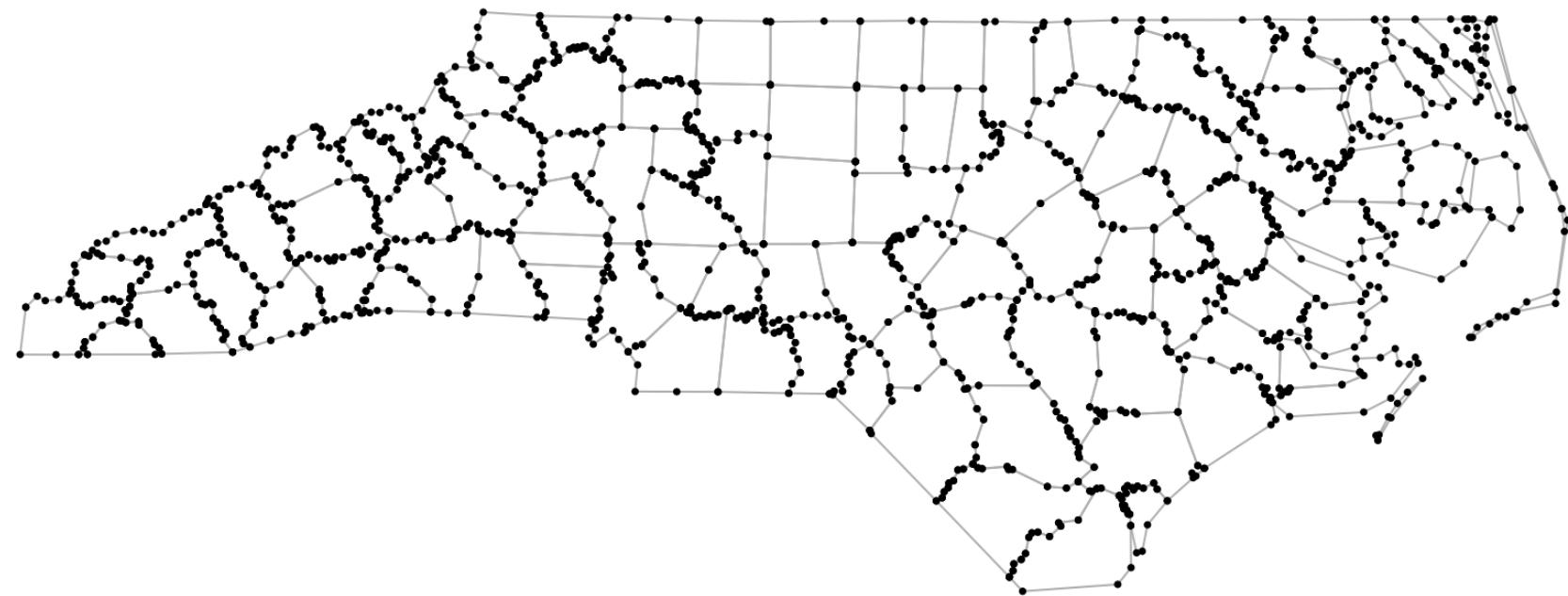
Bounding box: xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax:  
36.58965

Geodetic CRS: NAD27

# A tibble: 2,529 × 8

	NAME	BIR74	SID74	NWBIR74	BIR79	SID79	NWBIR79
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	Ashe	1091	1	10	1364	0	19
2	Ashe	1091	1	10	1364	0	19
3	Ashe	1091	1	10	1364	0	19

```
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

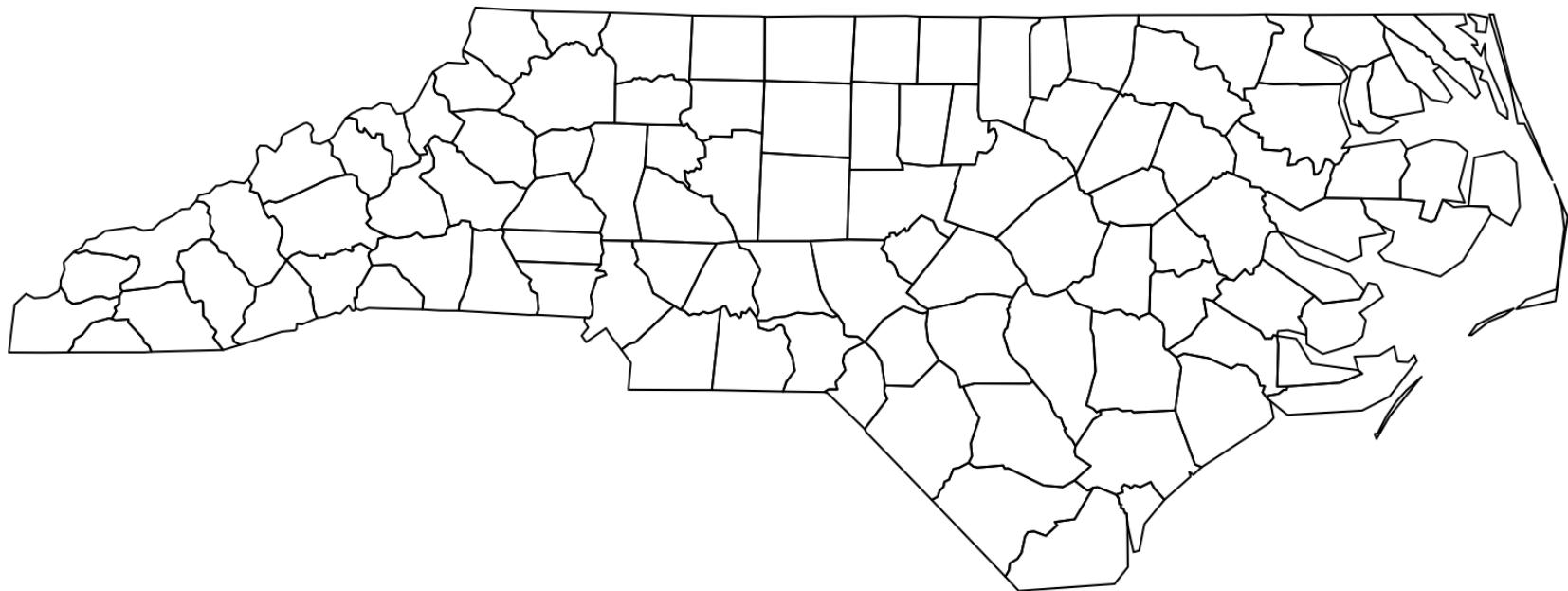
Bounding box: xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax:  
36.58965

Geodetic CRS: NAD27

# A tibble: 100 × 8

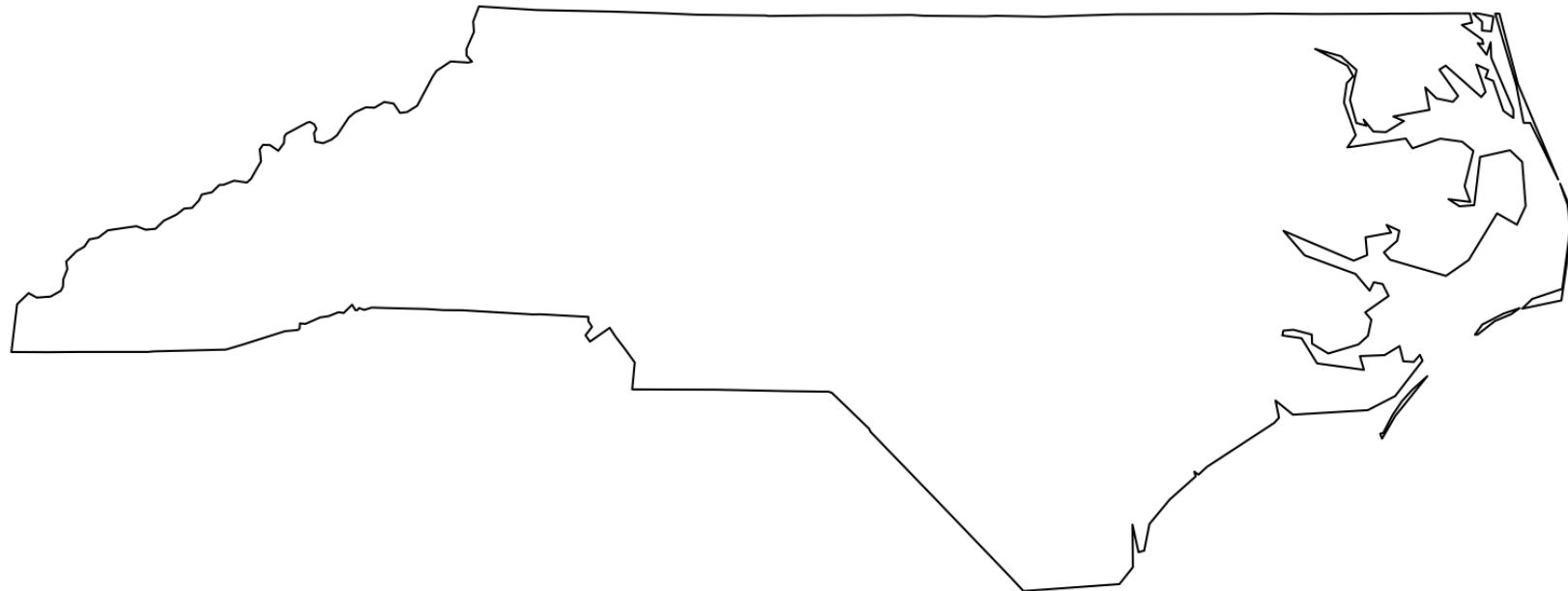
	NAME	BIR74	SID74	NWBIR74	BIR79	SID79	NWBIR79
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	Ashe	1091	1	10	1364	0	19
2	Alleghany	487	0	10	542	3	12
3	Surry	3188	5	208	3616	6	260

```
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

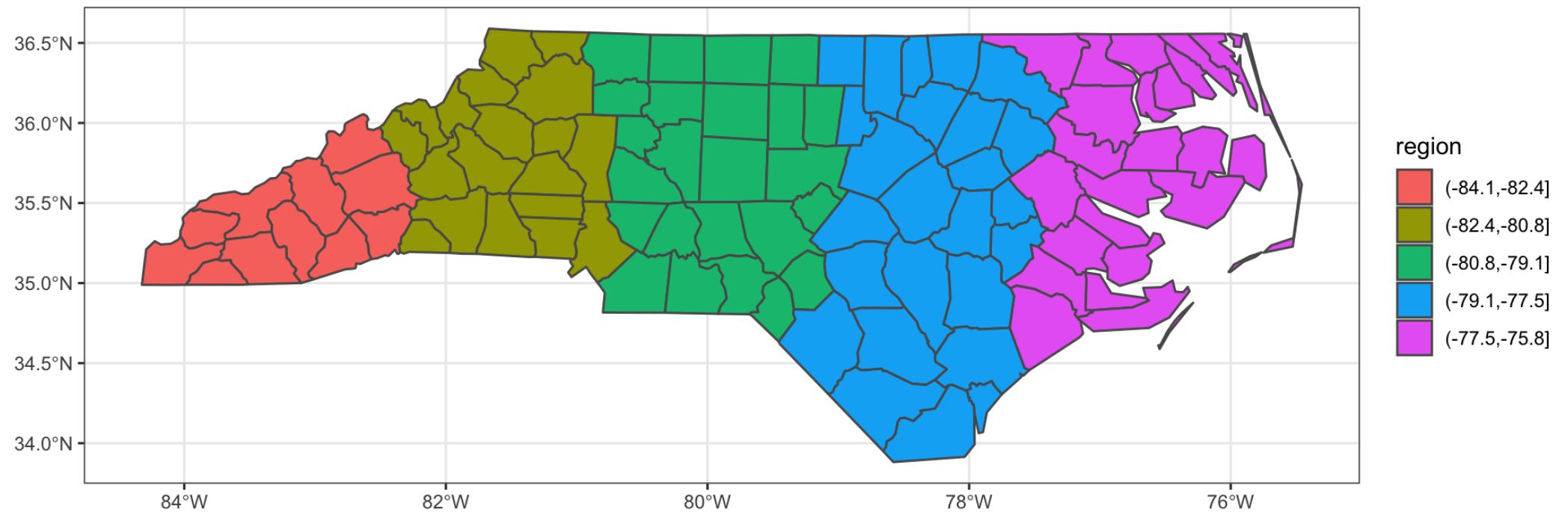
Bounding box: xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax:  
36.58965

Geodetic CRS: NAD27

# A tibble: 100 × 10

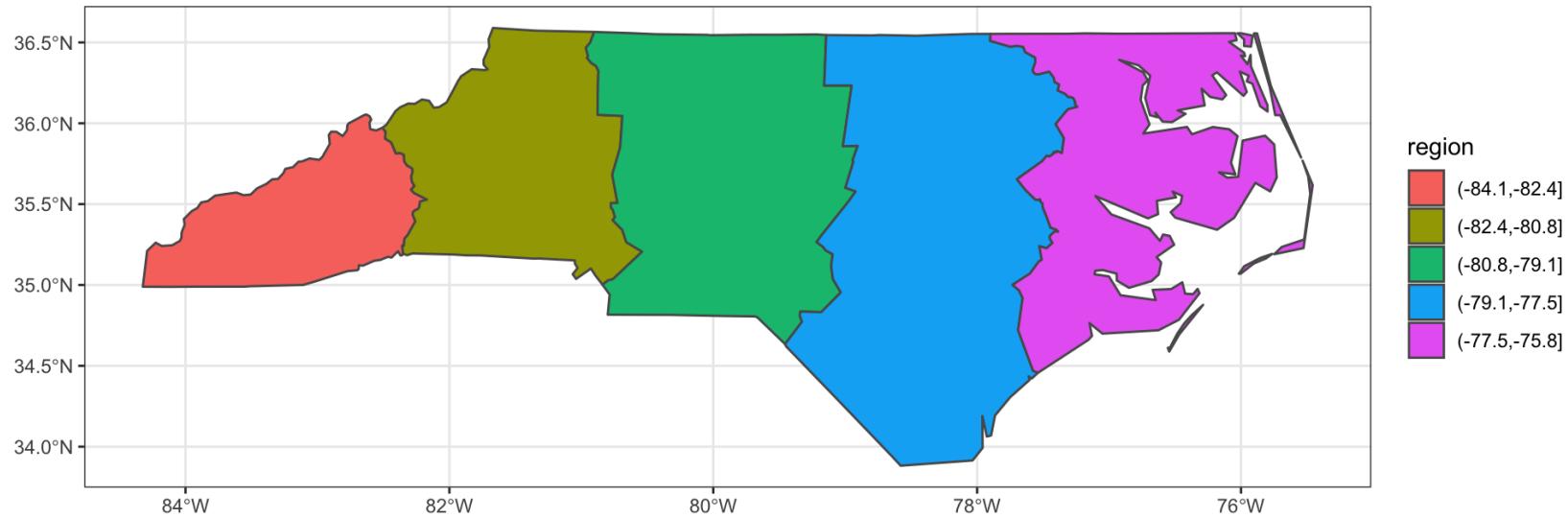
	NAME	BIR74	SID74	NWBIR74	BIR79	SID79	NWBIR79
*	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	Ashe	1091	1	10	1364	0	19
2	Alleghany	487	0	10	542	3	12
3	Surry	3188	5	208	3616	6	260

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



# dplyr and sf

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



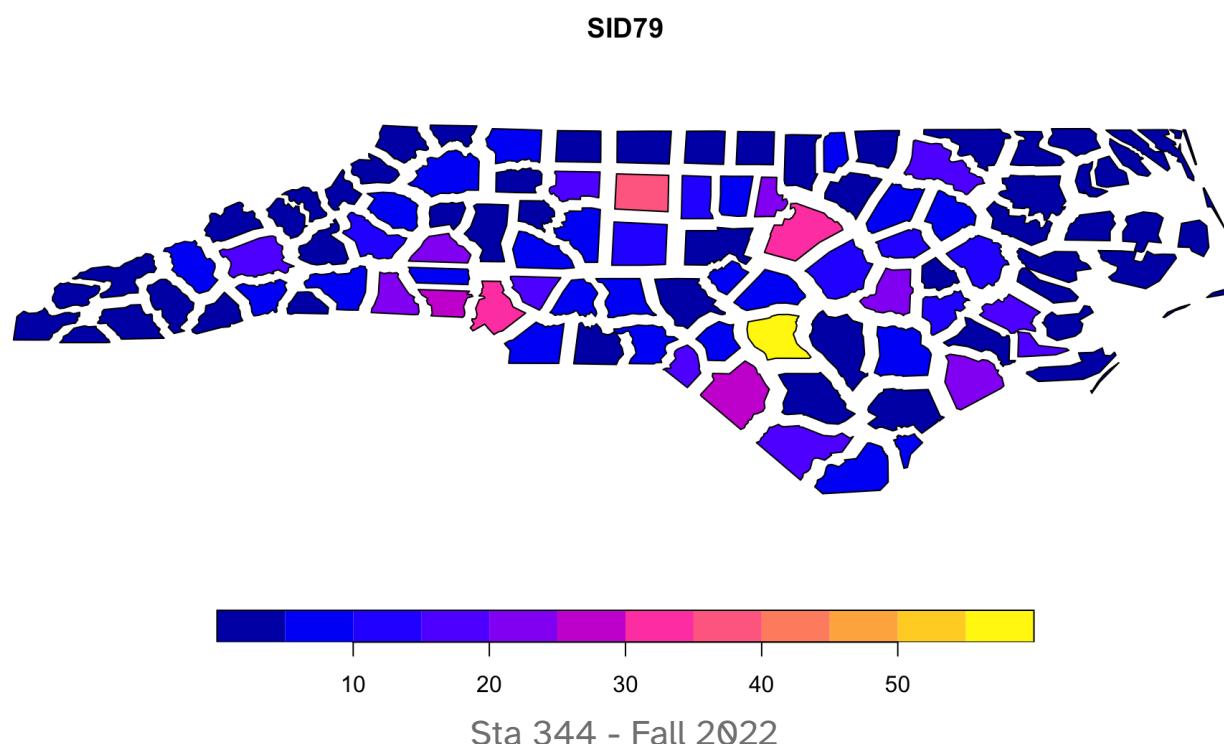
# Affine Transformations

```
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

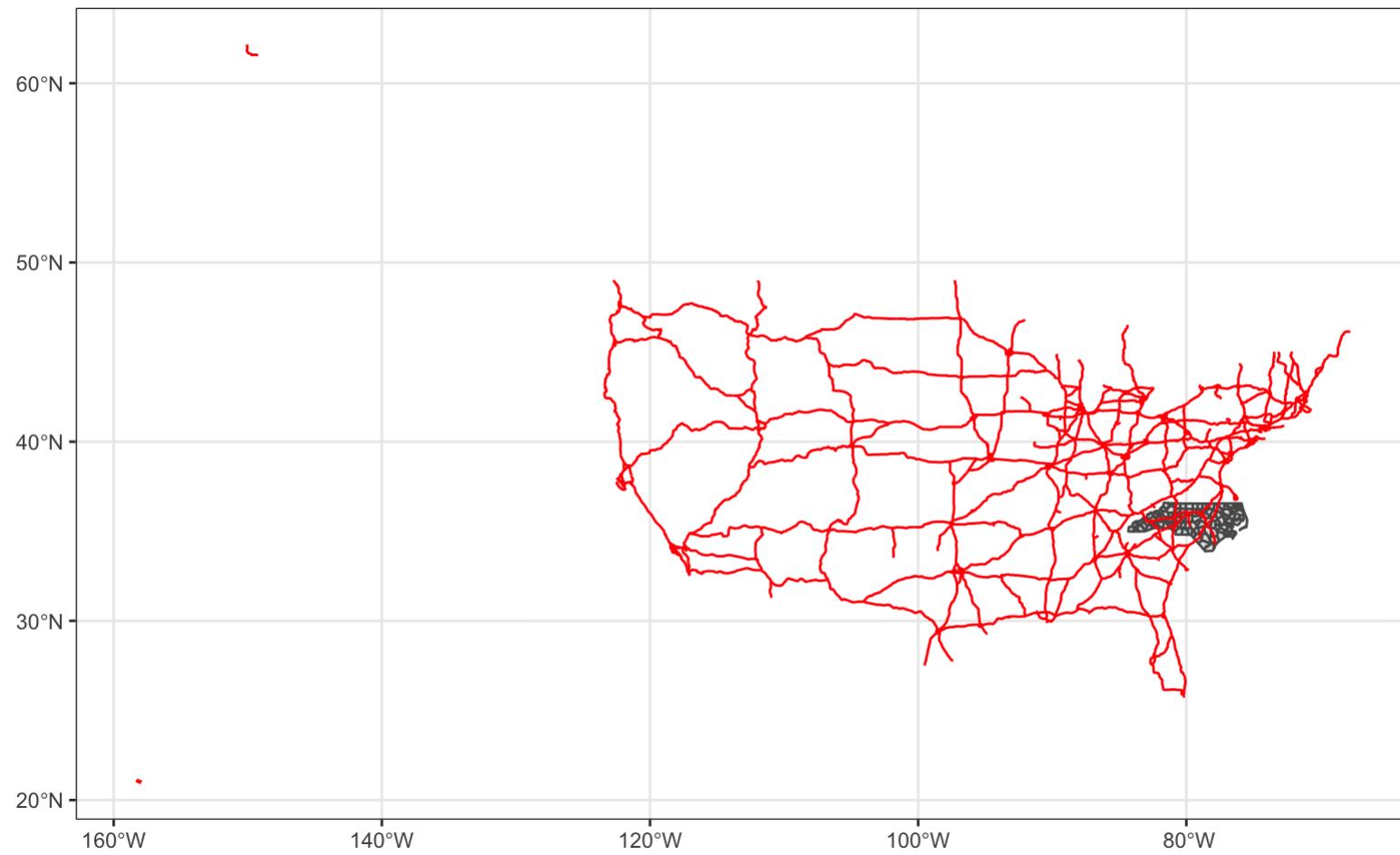
```
1 ctrd = st_centroid(st_geometry(nc))
2 area = st_area(nc) |> strip_attrs()
3
4 nc_rot = nc
5 st_geometry(nc_rot) = (st_geometry(nc) - ctrd) * 0.75 + ctrd
6
7 plot(nc_rot[, "SID79"])
```



# Highway Example

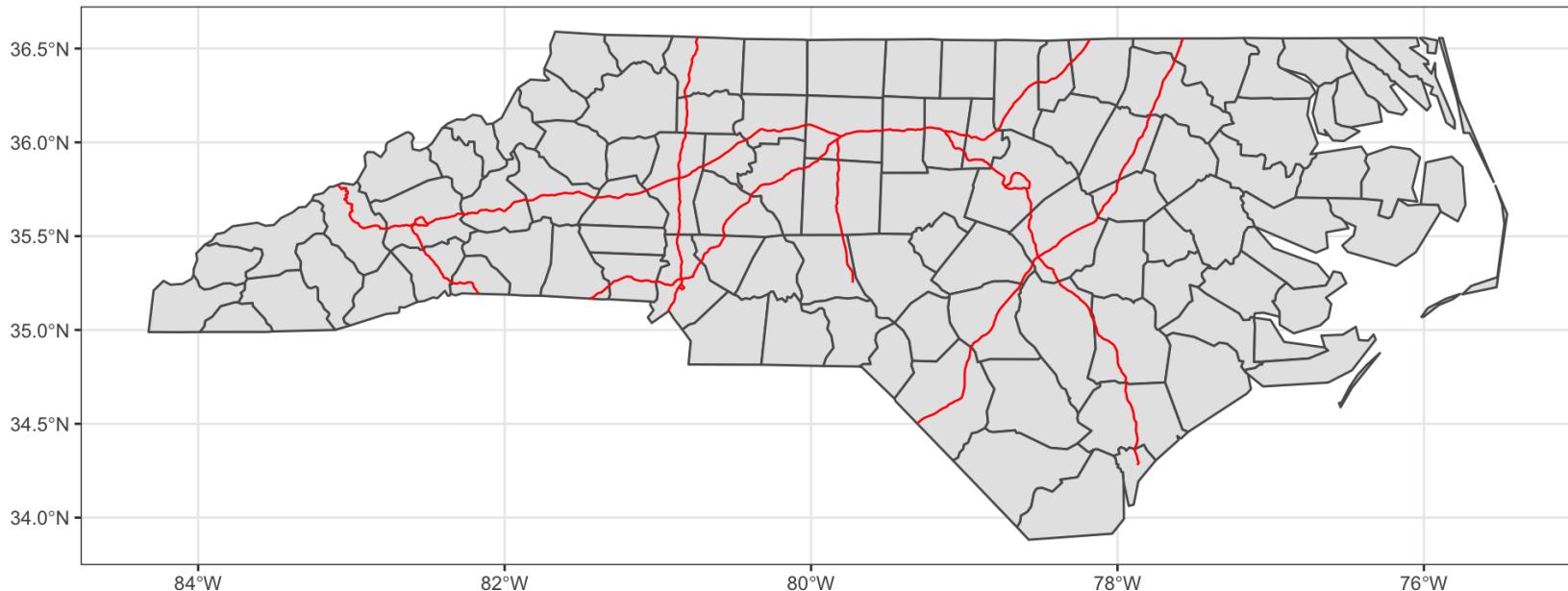
# Highways

```
1 hwy = read_sf("data/us_interstates.gpkg", quiet = TRUE) |>
2   st_transform(st_crs(nc))
3
4 ggplot() +
5   geom_sf(data=nc) +
6   geom_sf(data=hwy, col='red')
```



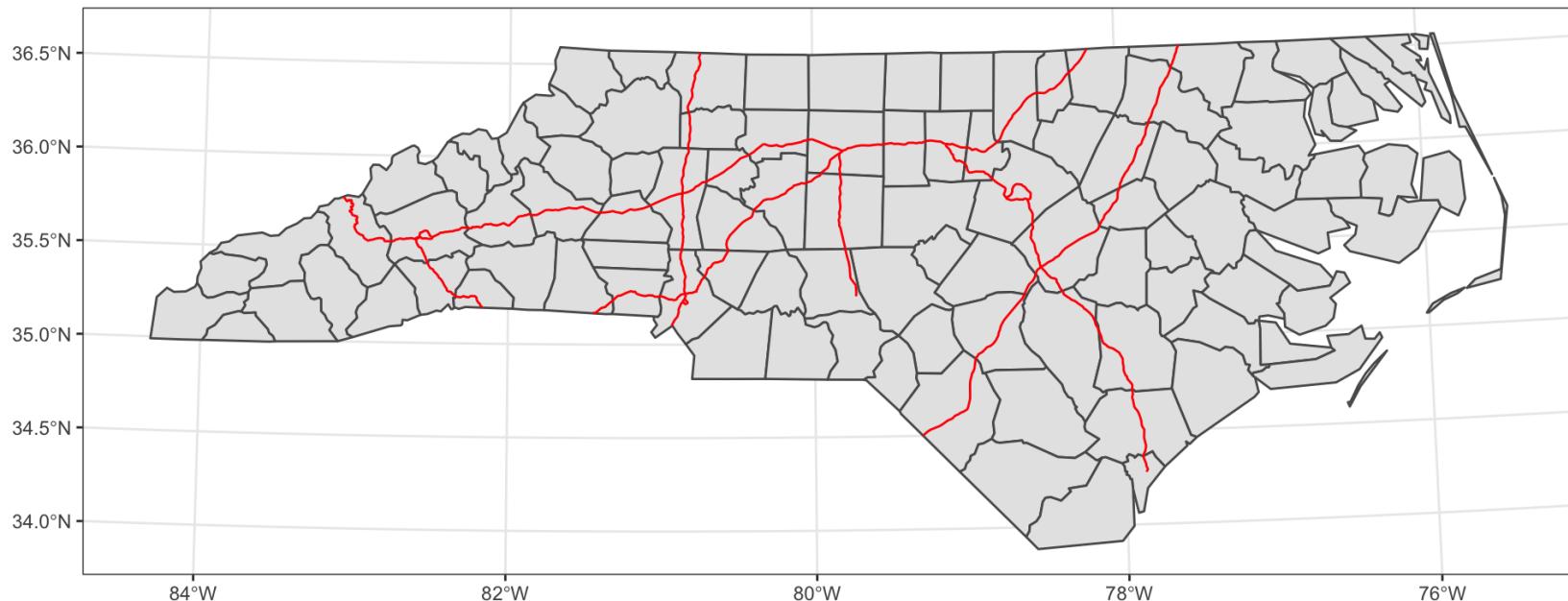
# NC Interstate Highways

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



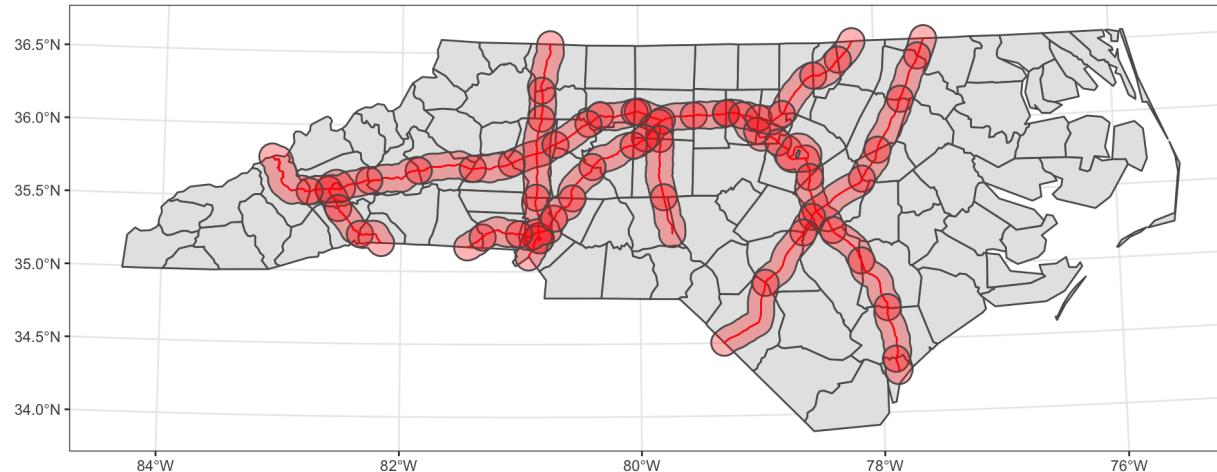
# Counties near the interstate (Projection)

```
1 nc_utm = st_transform(nc, "+proj=utm +zone=17 +datum=NAD83 +units=m +no_defs")
2
3 ggplot() +
4   geom_sf(data=nc_utm) +
5   geom_sf(data=hwy_nc, col='red')
```



# Counties near the interstate (Buffering)

```
1 hwy_nc_buffer = hwy_nc |>
2   st_transform("+proj=utm +zone=17 +datum=NAD83 +units=m +no_defs") |>
3   st_buffer(10000)
4
5 ggplot() +
6   geom_sf(data=nc_utm) +
7   geom_sf(data=hwy_nc, color='red') +
8   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_transform("+proj=utm +zone=17 +datum=NAD83 +units=m +no_defs") |>
3   st_buffer(10000) |>
4   st_union() |>
5   st_sf()
6
7 ggplot() +
8   geom_sf(data=nc_utm) +
9   geom_sf(data=hwy_nc, color='red') +
10  geom_sf(data=hwy_nc_buffer, fill='red', alpha=0.3)
```

# Example

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

# Gerrymandering Example

# NC House Districts - 112th Congress

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

Simple feature collection with 13 features and 2 fields

Geometry type: MULTIPOLYGON

Dimension: XY

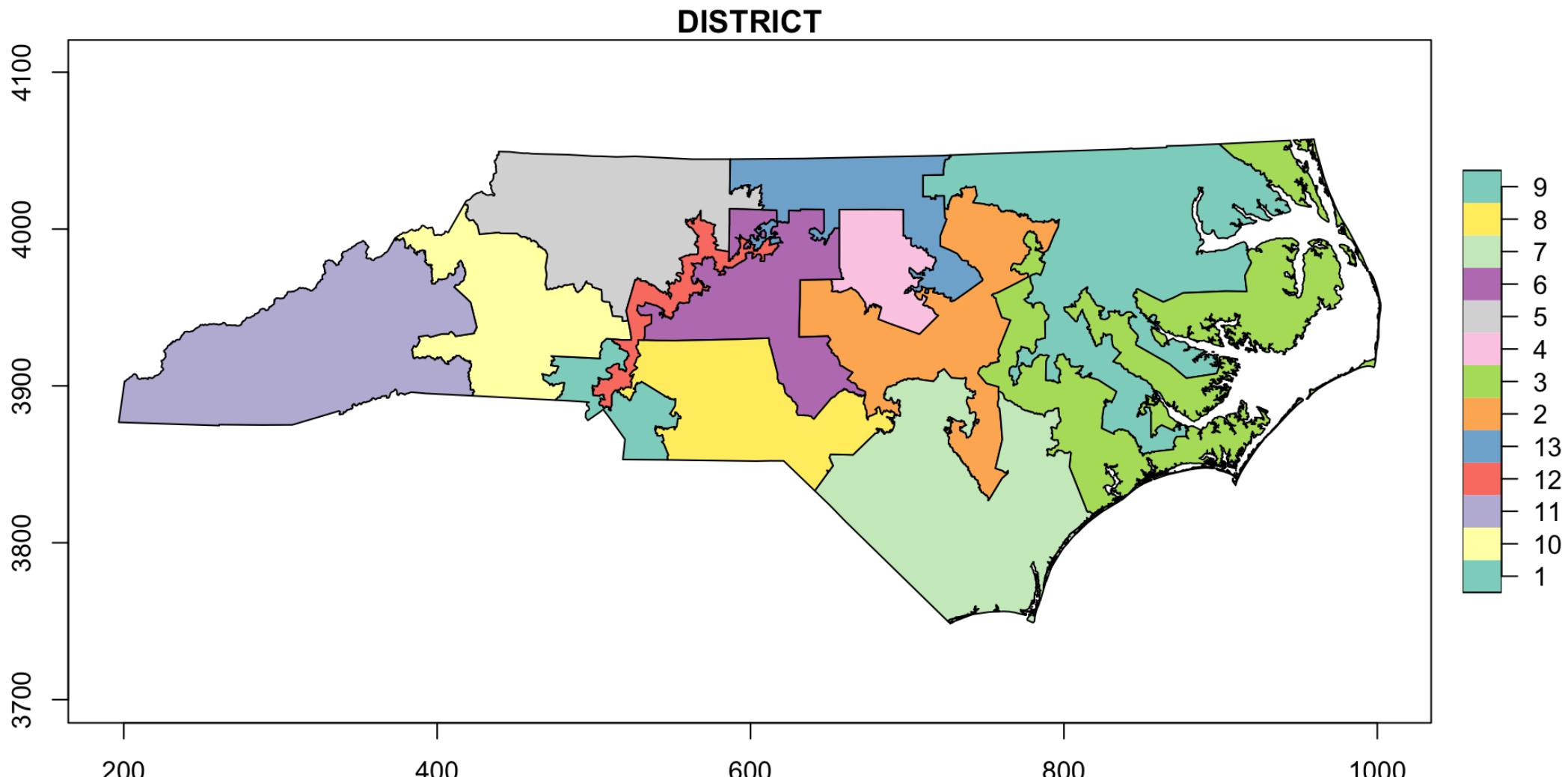
Bounding box: xmin: -84.32187 ymin: 33.84452 xmax: -75.45998 ymax: 36.58812

Geodetic CRS: WGS 84

# A tibble: 13 × 3

	ID	DISTRICT	geom
	<chr>	<chr>	<MULTIPOLYGON [°]>
1	037108112001	1	(((-77.32845 35.35031, -77.35398 3...
2	037108112002	2	(((-78.89928 35.12619, -78.89763 3...
3	037108112003	3	(((-75.68266 35.23291, -75.68113 3...
4	037108112004	4	(((-78.77926 35.78568, -78.77947 3...
5	037108112005	5	(((-79.8968 36.38075, -79.89213 36...

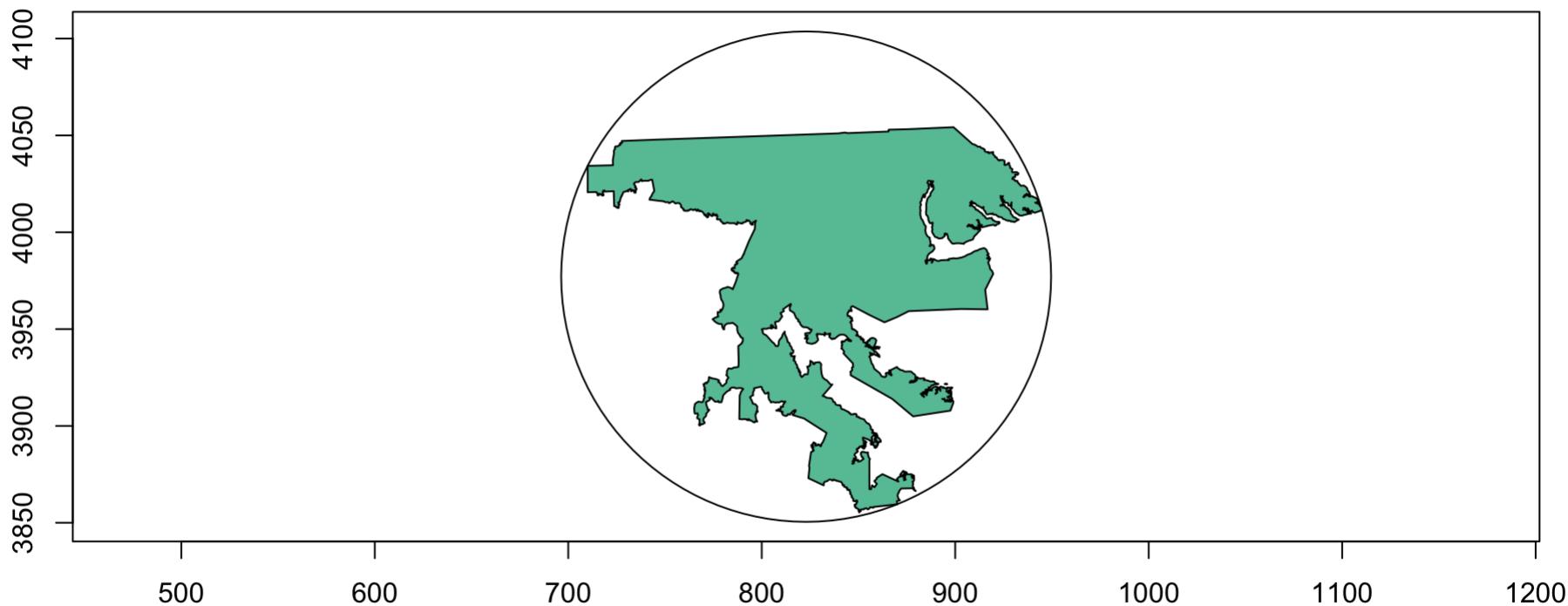
```
1 nc_house = nc_house |>  
2   st_transform("+proj=utm +zone=17 +datum=NAD83 +units=km +no_defs")  
3 plot(nc_house[, "DISTRICT"], axes=TRUE)
```



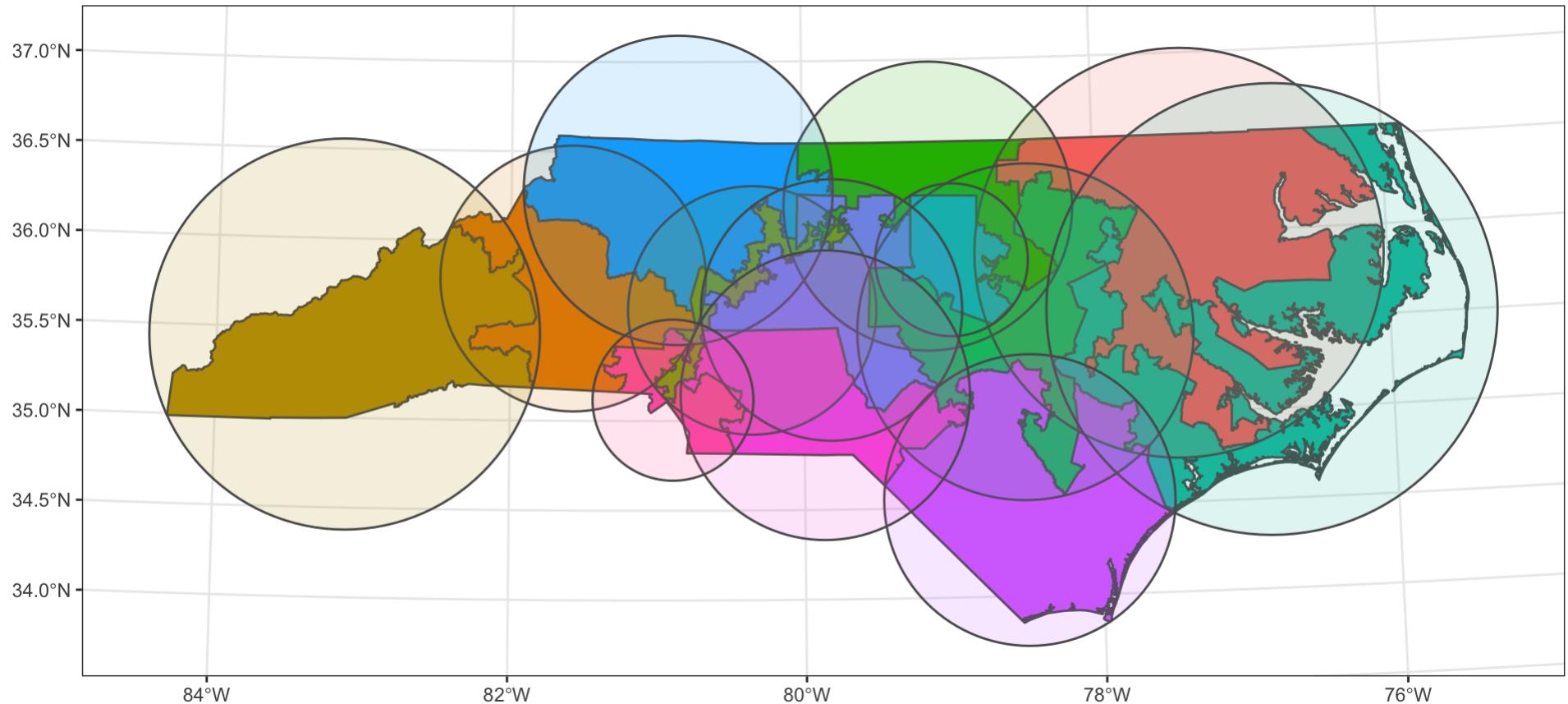
# Measuring Compactness - Reock Score

The Reock score is a measure of compactness that is calculated as the ratio 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)
```

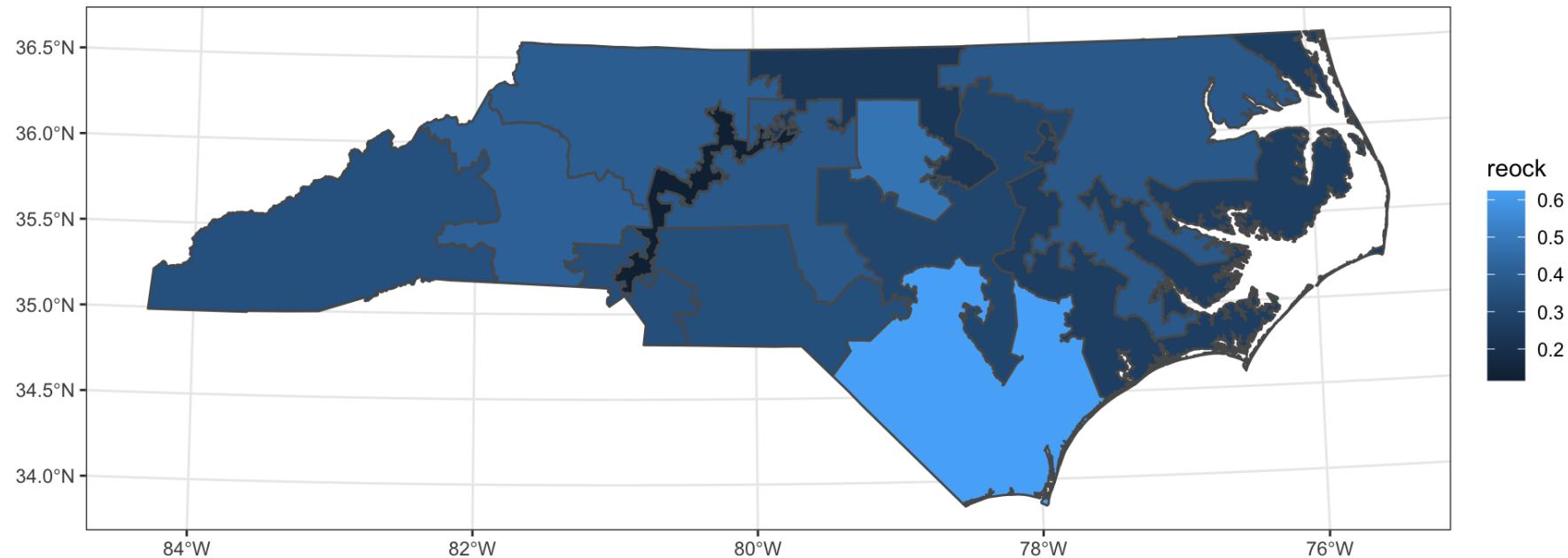


```
1 ggplot(mapping = aes(fill=DISTRICT)) +  
2   geom_sf(data=nc_house) +  
3   geom_sf(data=circs, alpha=0.15) +  
4   guides(color="none", fill="none")
```



# 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()
```



```

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: 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	DISTRICT	geom	reock
	<chr>	<chr>	<MULTIPOLYGON [km]>	[1]
1	037108112012	12	((545.2987 3945.168, 543.52...	0.116
2	037108112013	13	((597.9665 4026.852, 598.03...	0.237
3	037108112003	3	((984.0709 3911.853, 984.21...	0.266
4	037108112002	2	((691.4141 3889.056, 691.55...	0.303
5	037108112009	9	((506.3207 3893.208, 506.88...	0.339
6	037108112008	8	((688.6584 3870.456, 688.02...	0.342
7	037108112011	11	((226.7522 3918.52, 226.716...	0.344
8	037108112006	6	((552.4691 3949.669, 552.94...	0.378

# 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.
test.grd	138.7071	293.9575	371.9001	425.606	501.0102
	Max.	NA's			

test.grd 1736.058 6022

dimension(s):

	from	to	offset	delta	refsys	point
x	1	80	NA	NA	Amersfoort / RD New	NA
y	1	115	NA	NA	Amersfoort / RD New	NA
				values x/y		

x [80x115] 178451,...,181611 [x]

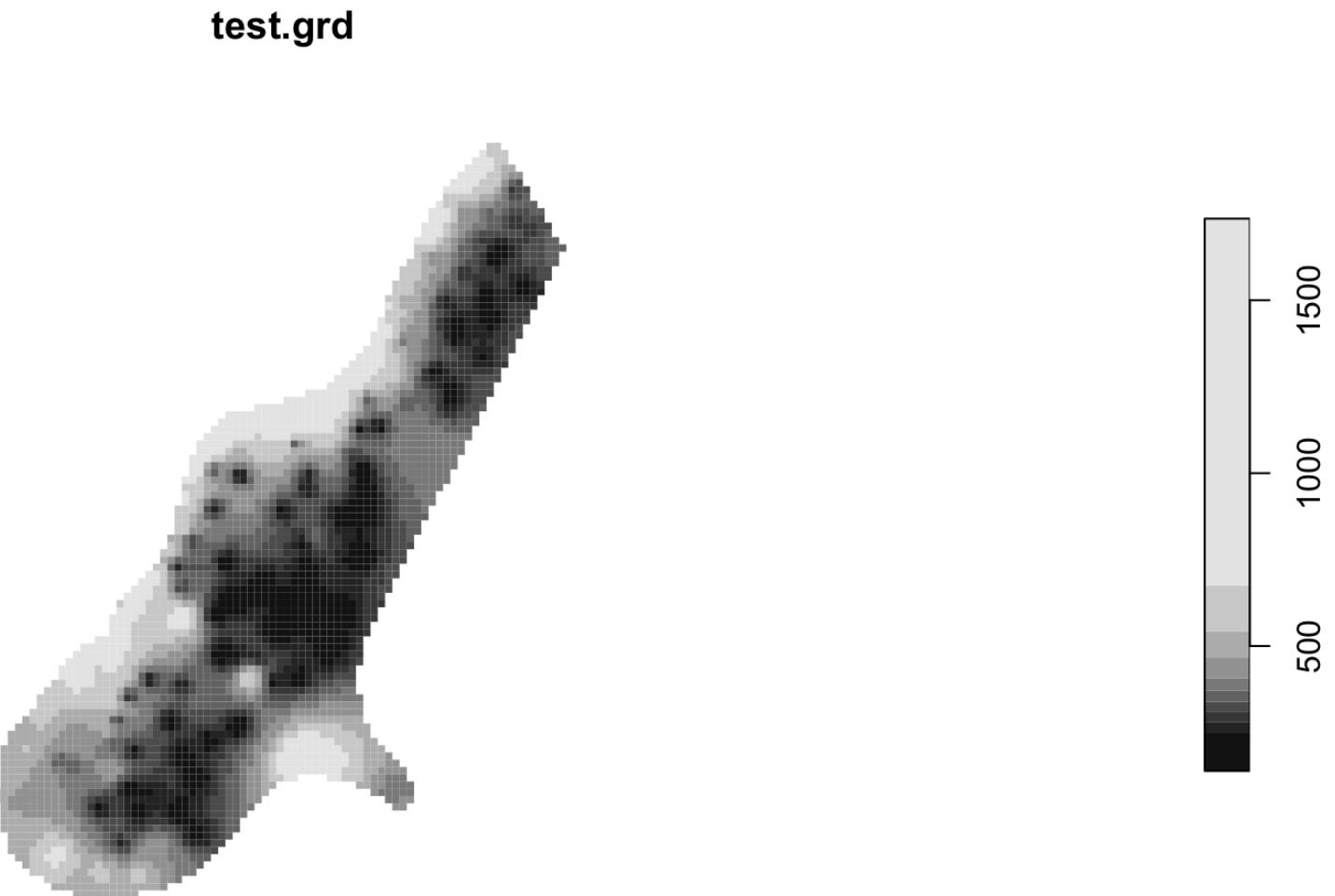
# stars class

```
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"
```

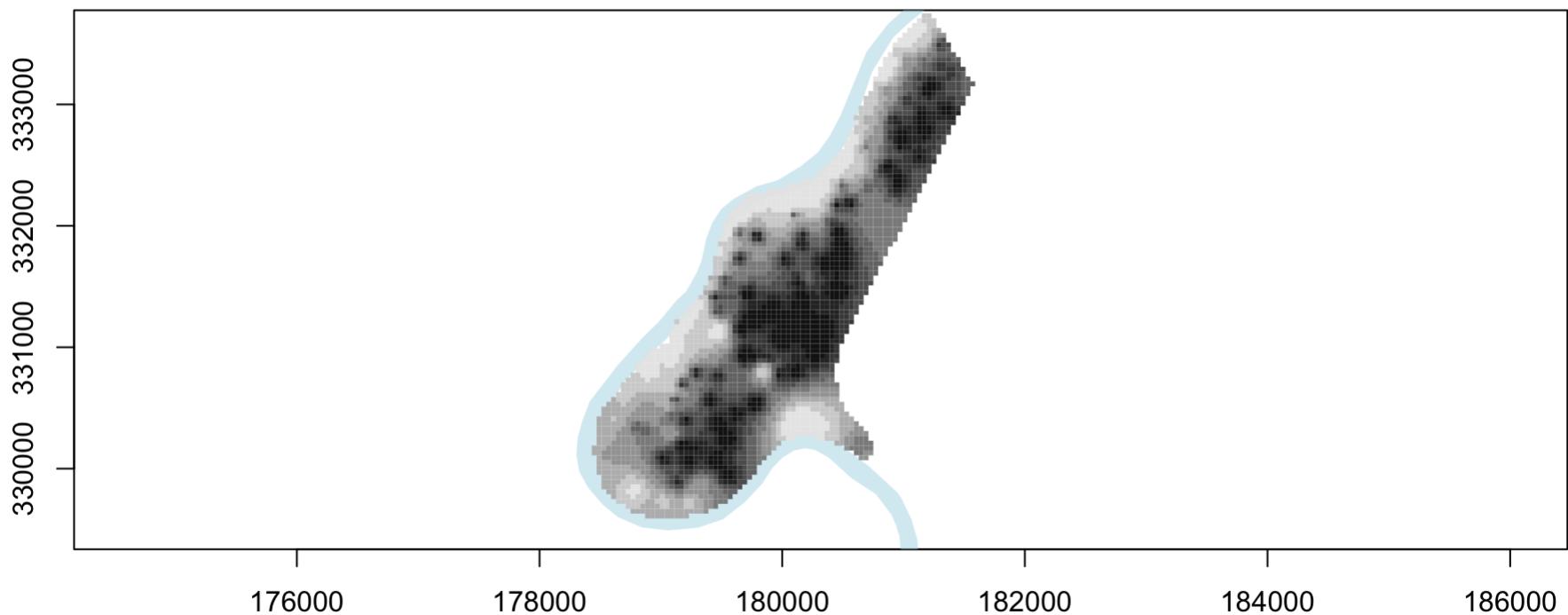
# Plotting

```
1 plot(meuse_rast)
```



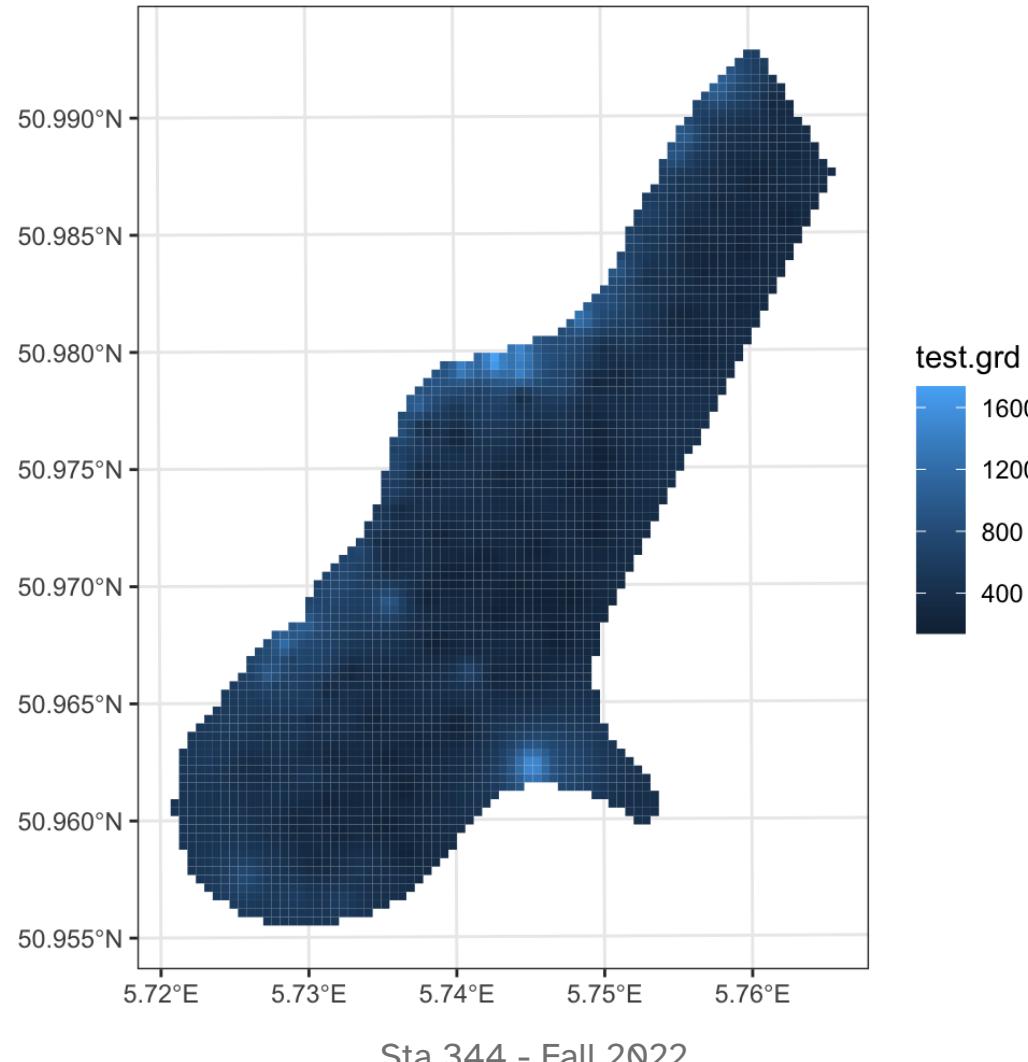
```
1 plot(  
2   meuse_riv,  
3   col=adjustcolor("lightblue",alpha.f = 0.5), border=NA,  
4   ylim = c(329500, 333611), axes=TRUE  
5 )  
6 plot(meuse_rast, add=TRUE)
```

test.grd

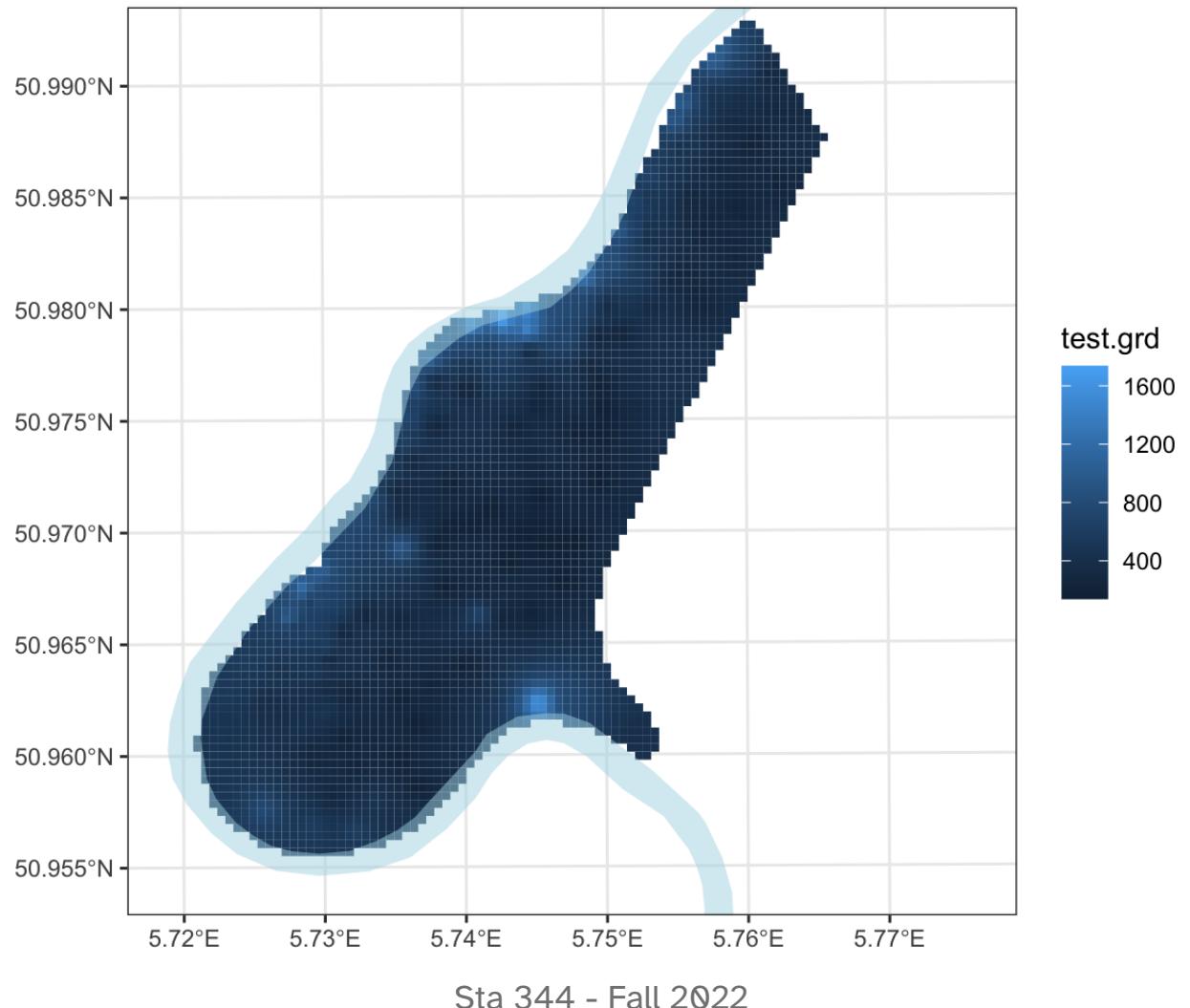


# ggplot

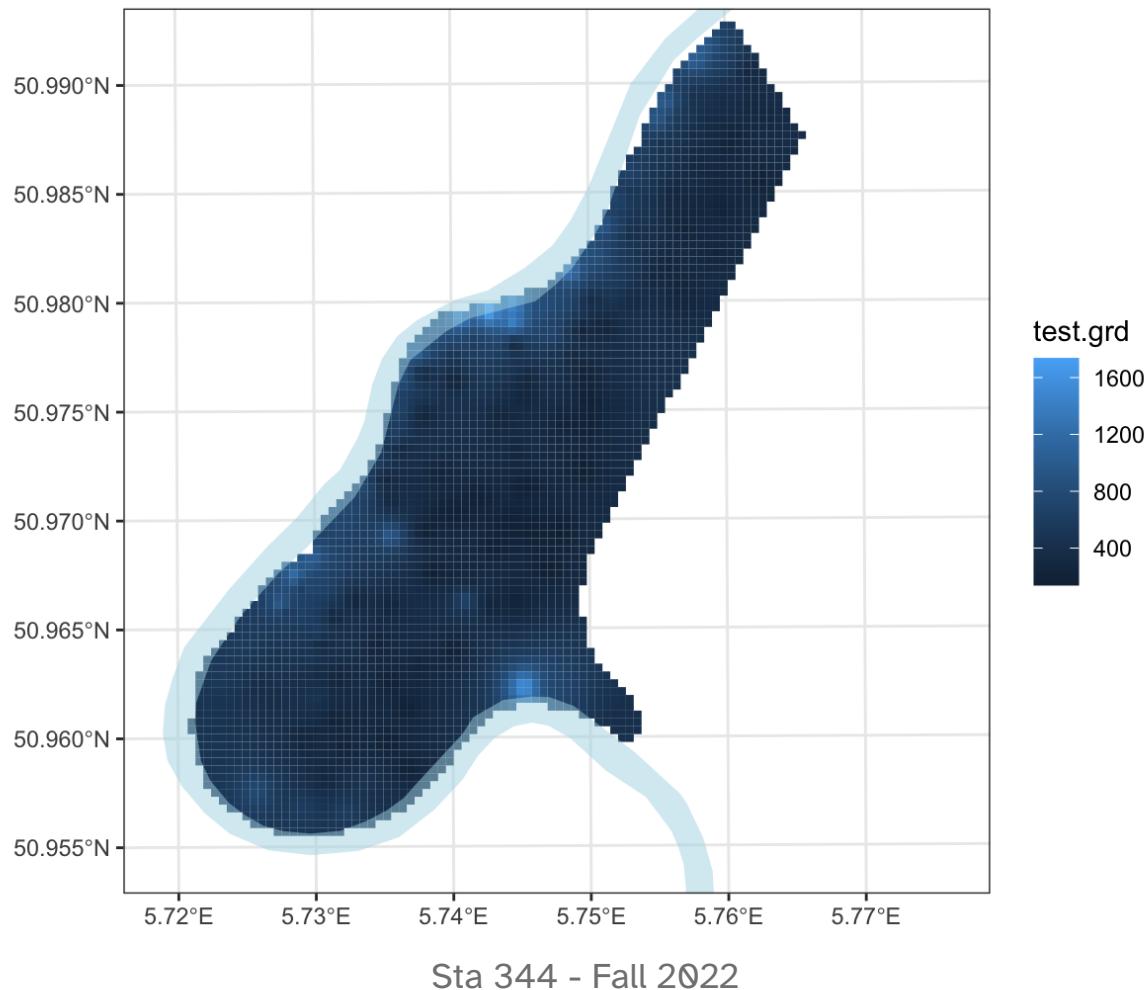
```
1 ggplot() +  
2   stars::geom_stars(data=meuse_rast)
```



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



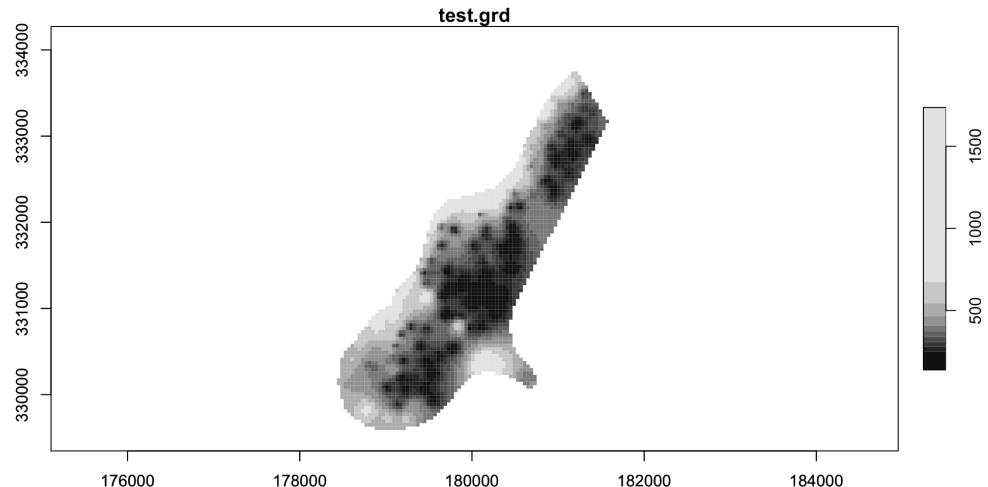
```
1 ggplot() +  
2   stars::geom_stars(data=meuse_rast) +  
3   geom_sf(data=meuse_riv, fill="lightblue", color=NA, alpha=0.5) +  
4   ylim(329500, 333611) +  
5   scale_fill_gradient(na.value = NA)
```



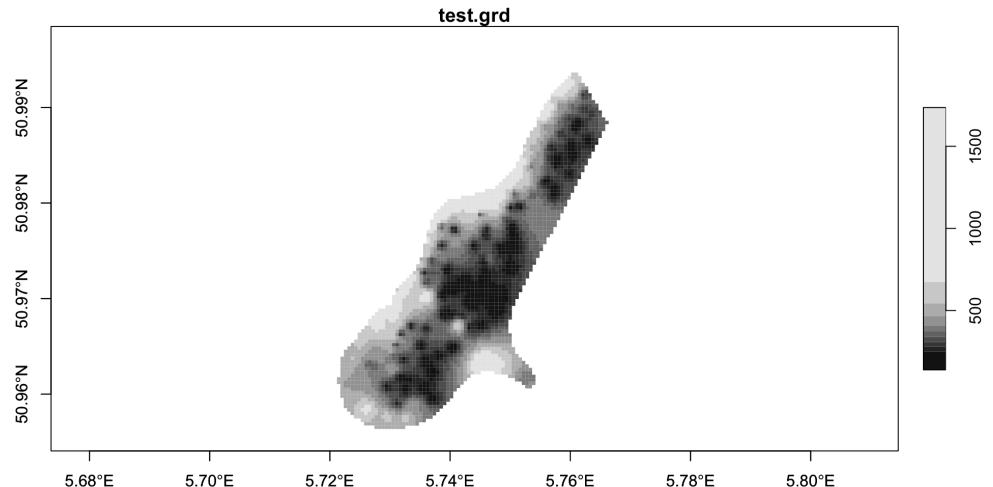
# Rasters and Projections

```
1 meuse_rast_ll = st_transform(meuse_rast, "+proj=longlat +datum=NAD83 +no_defs")
```

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



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



```
1 meuse_rast
```

stars object with 2 dimensions and 1 attribute  
attribute(s):

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

dimension(s):

	from	to	offset	delta	refsys
point					
x	1	80	NA	NA	Amersfoort / RD New
NA					
y	1	115	NA	NA	Amersfoort / RD New
NA					
					values x/v

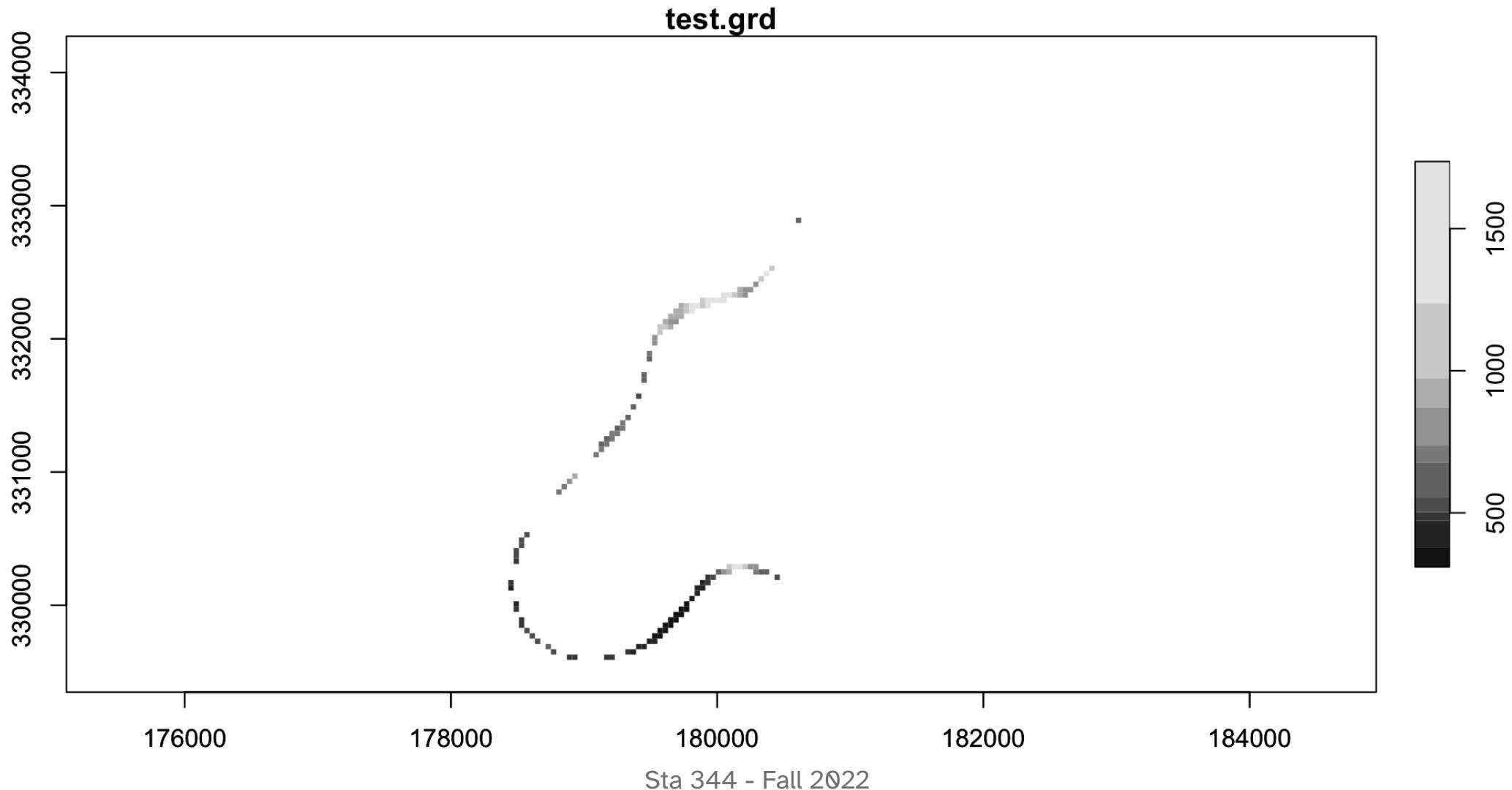
```
1 meuse_rast_ll
```

stars object with 2 dimensions and 1 attribute  
attribute(s):

	Min.	1st Qu.	Median	Mean	
3rd Qu.					
test.grd	138.7071	293.9575	371.9001	425.606	
	501.0102				
	Max.	NA's			
test.grd	1736.058	6022			
	dimension(s):				
	from	to	offset	delta	
	refsys	point			
x	1	80	NA	NA	+proj=longlat
	+datum=NAD8...				NA
y	1	115	NA	NA	+proj=longlat
	+datum=NAD8...				NA
					values x/v

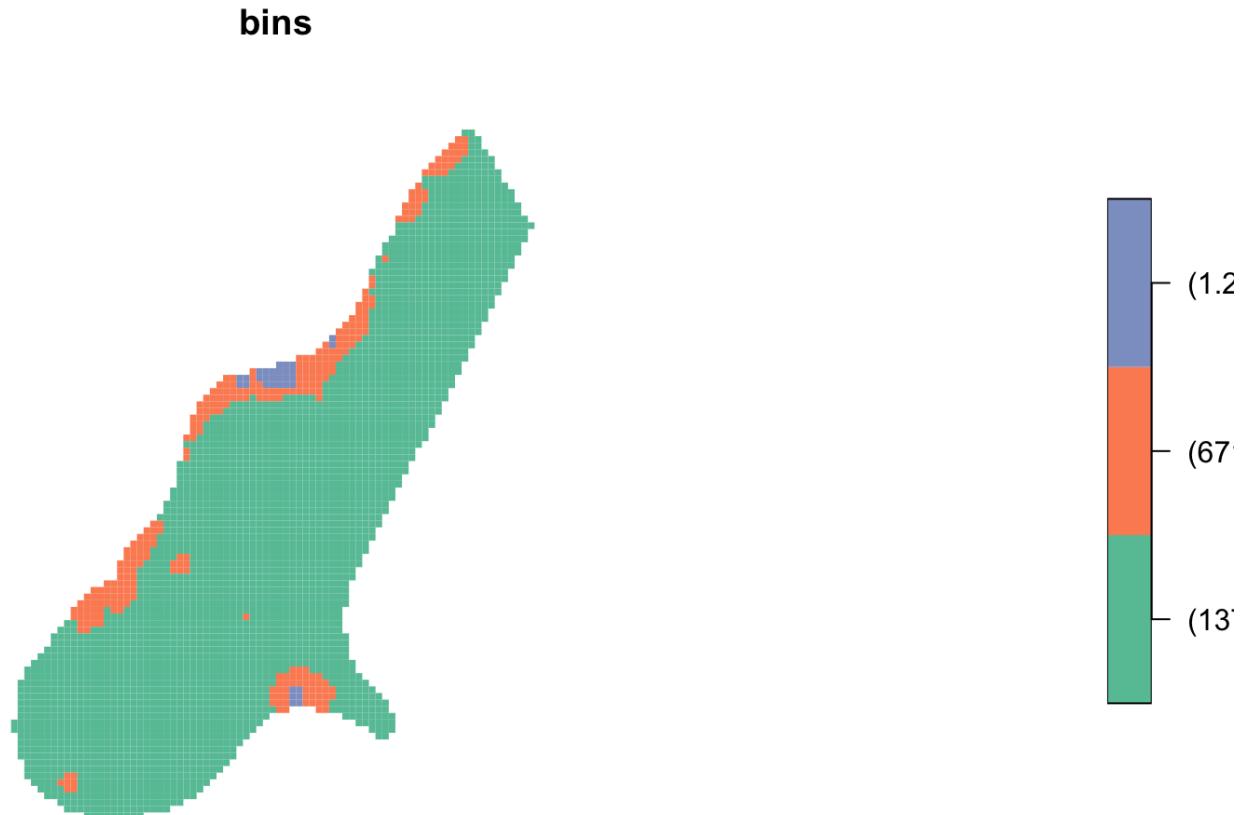
# Cropping

```
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()
```

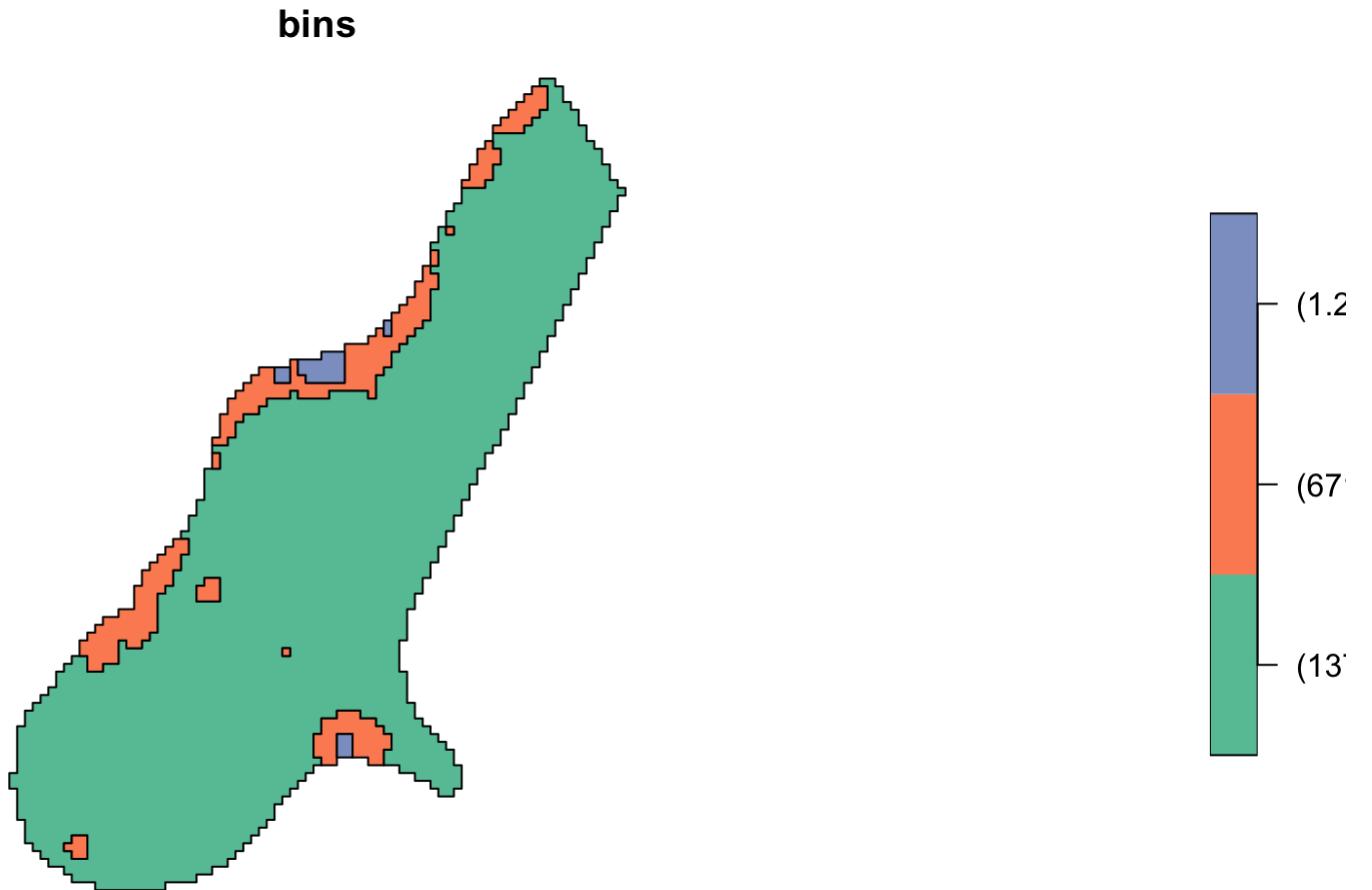


# Polygonization

```
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 <sup>2</sup> ]
1	(137,671]	POLYGON ((178551 329829.8, 1785...	4.56e6
2	(671,1.2e+03]	MULTIPOLYGON (((178711 329829.8...	4.74e5
3	(1.2e+03,1.74e+03]	MULTIPOLYGON (((179790.9 332189...	5.12e4