Lecture 16

Spatial Data and Cartography (Part 2)

3/22/2018

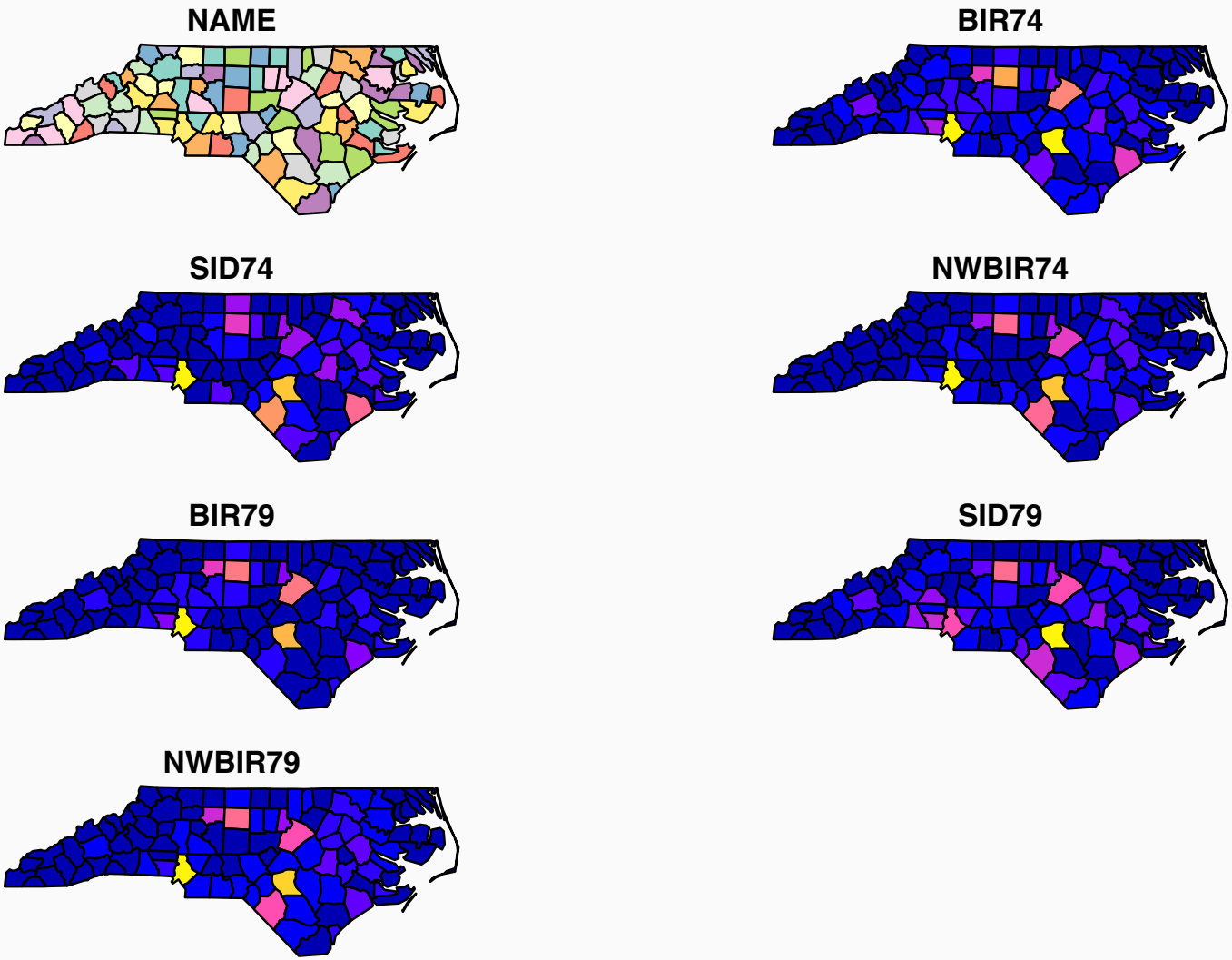
Plotting

Example Data - NC SIDS

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

tbl_df(nc)
## # A tibble: 100 x 8
##    NAME  BIR74 SID74 NWBIR74 BIR79 SID79 NWBIR79                  geometry
##    <fct> <dbl> <dbl>   <dbl> <dbl> <dbl>   <dbl>    <sf_geometry [degree]>
##  1 Ashe  1091.    1.     10. 1364.    0.     19. MULTIPOLYGON (((-81.4727~
##  2 Alle~  487.    0.     10.  542.    3.     12. MULTIPOLYGON (((-81.2398~
##  3 Surry 3188.    5.    208. 3616.    6.    260. MULTIPOLYGON (((-80.4563~
##  4 Curr~  508.    1.    123.  830.    2.    145. MULTIPOLYGON (((-76.0089~
##  5 Nort~ 1421.    9.   1066. 1606.    3.   1197. MULTIPOLYGON (((-77.2176~
##  6 Hert~ 1452.    7.    954. 1838.    5.   1237. MULTIPOLYGON (((-76.7450~
##  7 Camd~  286.    0.    115.  350.    2.    139. MULTIPOLYGON (((-76.0089~
##  8 Gates  420.    0.    254.  594.    2.    371. MULTIPOLYGON (((-76.5625~
##  9 Warr~  968.    4.    748. 1190.    2.    844. MULTIPOLYGON (((-78.3087~
## 10 Stok~ 1612.    1.    160. 2038.    5.    176. MULTIPOLYGON (((-80.0256~
## # ... with 90 more rows
```

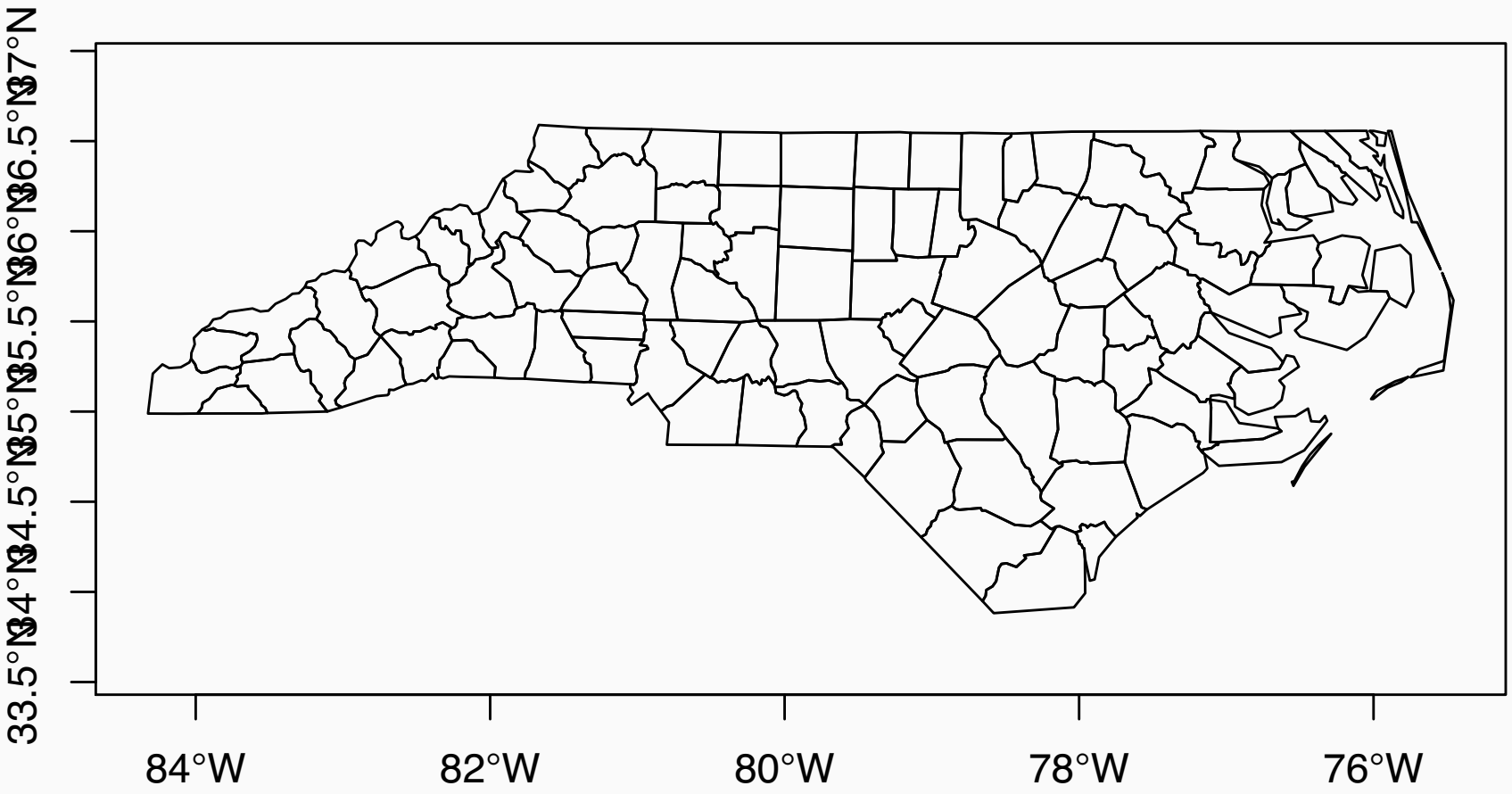
Base Plots

```
plot(nc)
```



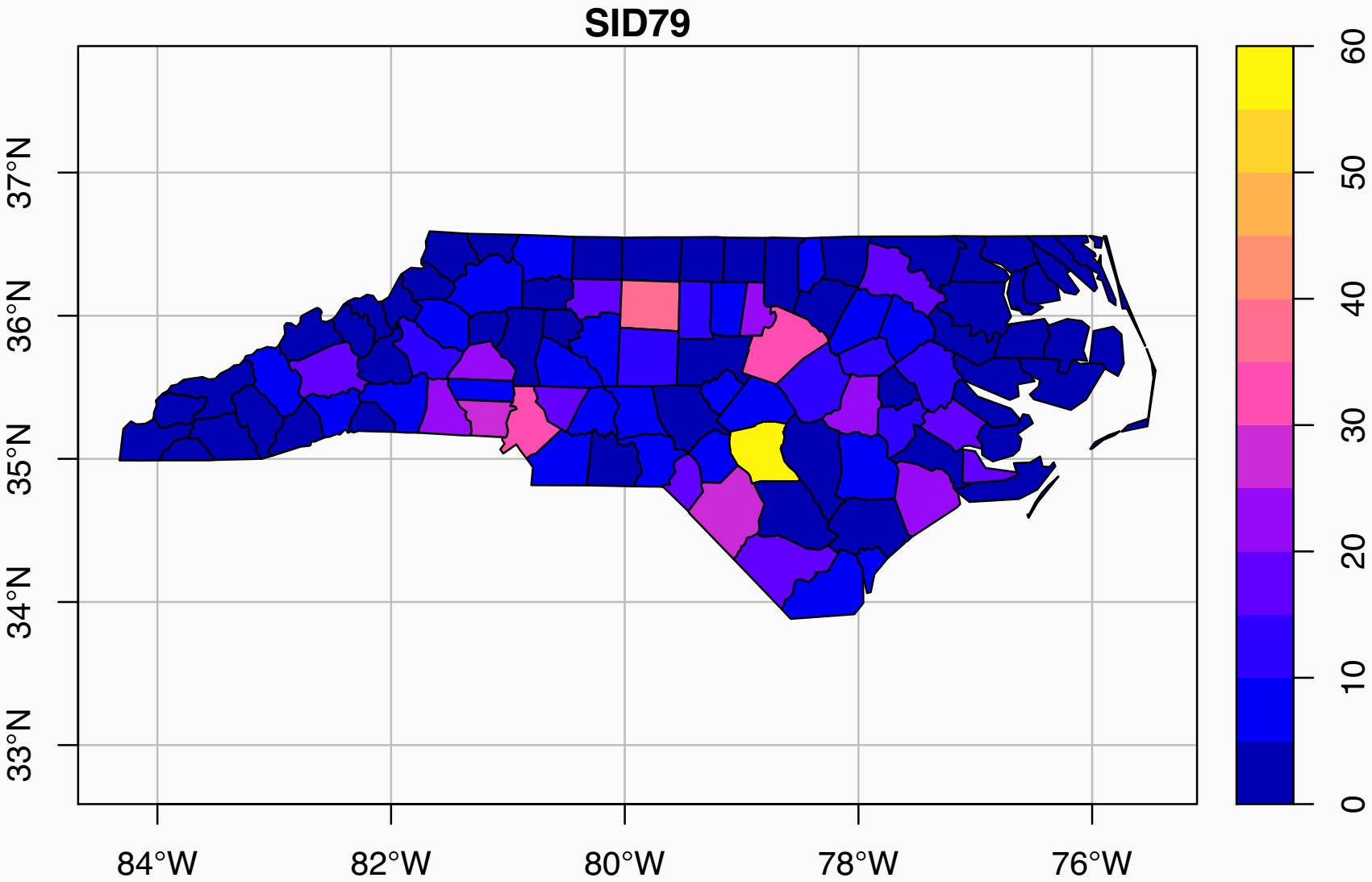
Geometry Plot

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



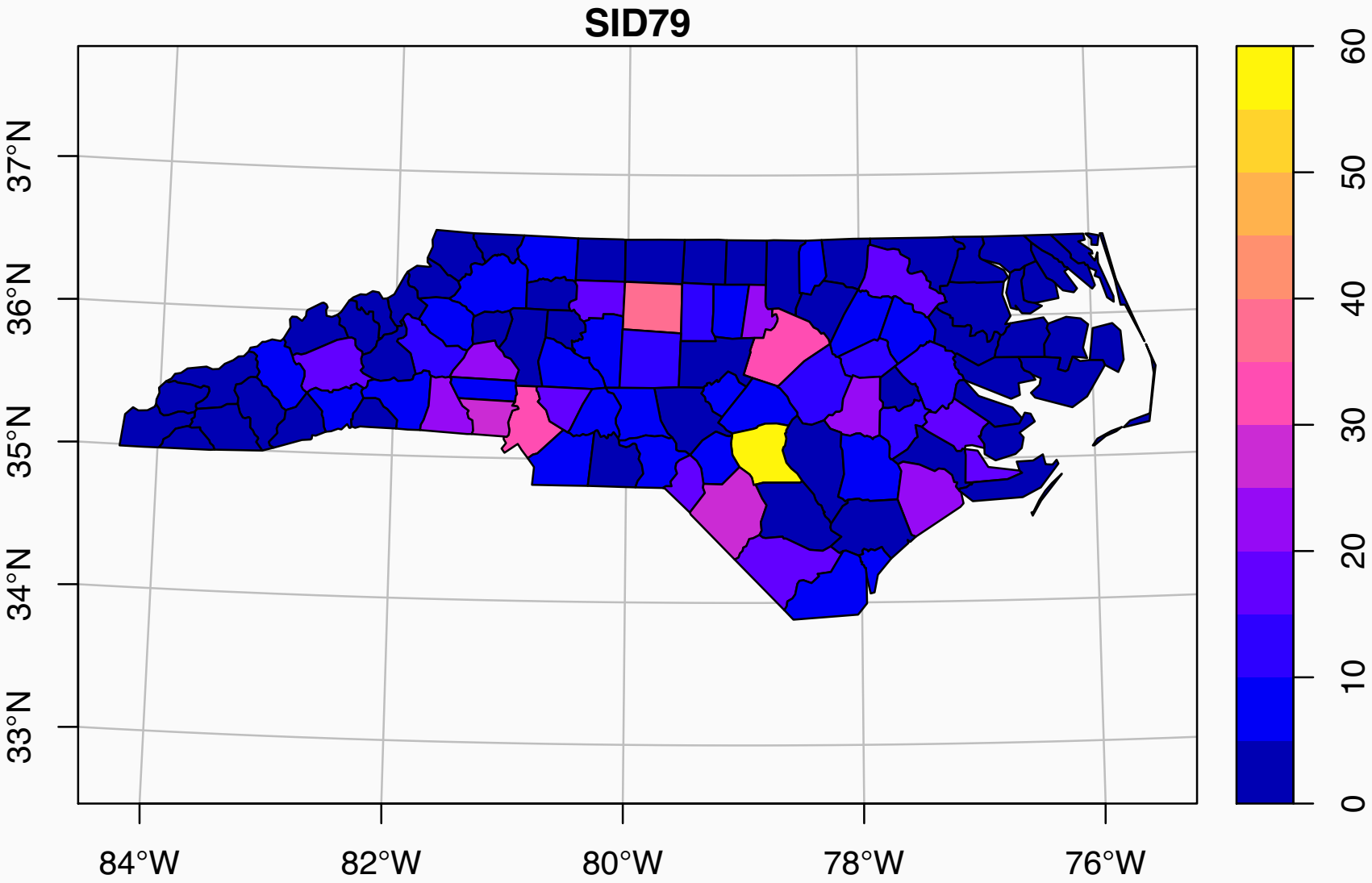
Graticules

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



Graticules (EPSG:3631)

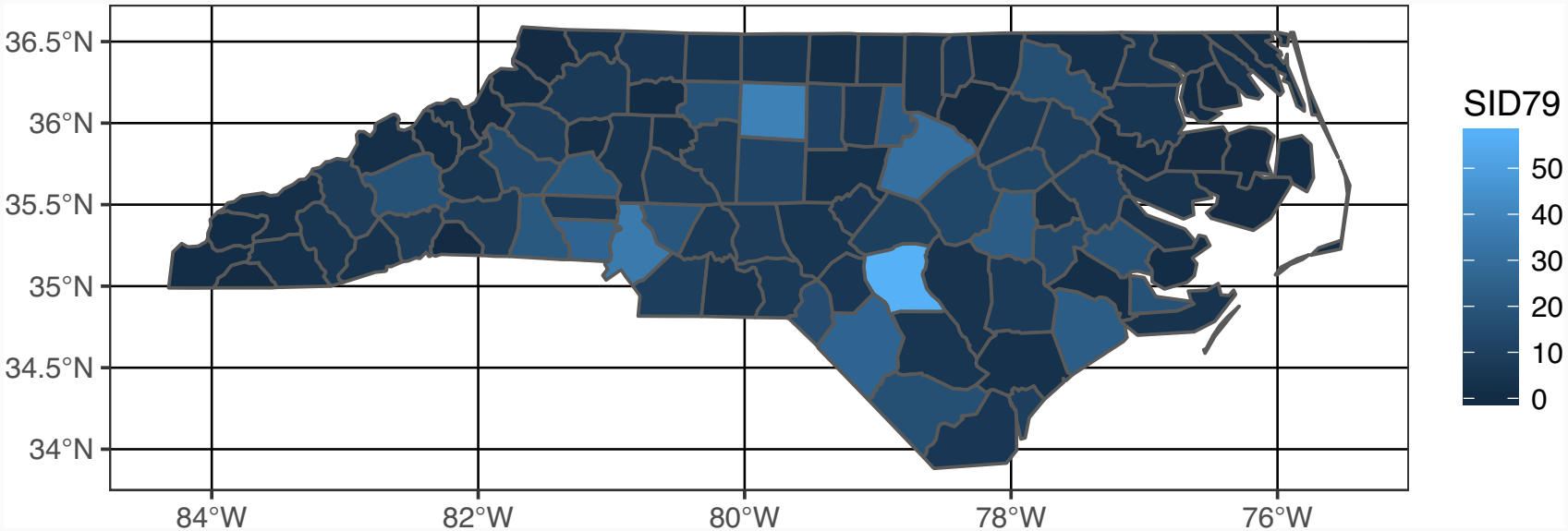
```
plot(st_transform(nc[,"SID79"], 3631), graticule=st_crs(nc), axes=TRUE)
```



ggplot2 (dev)

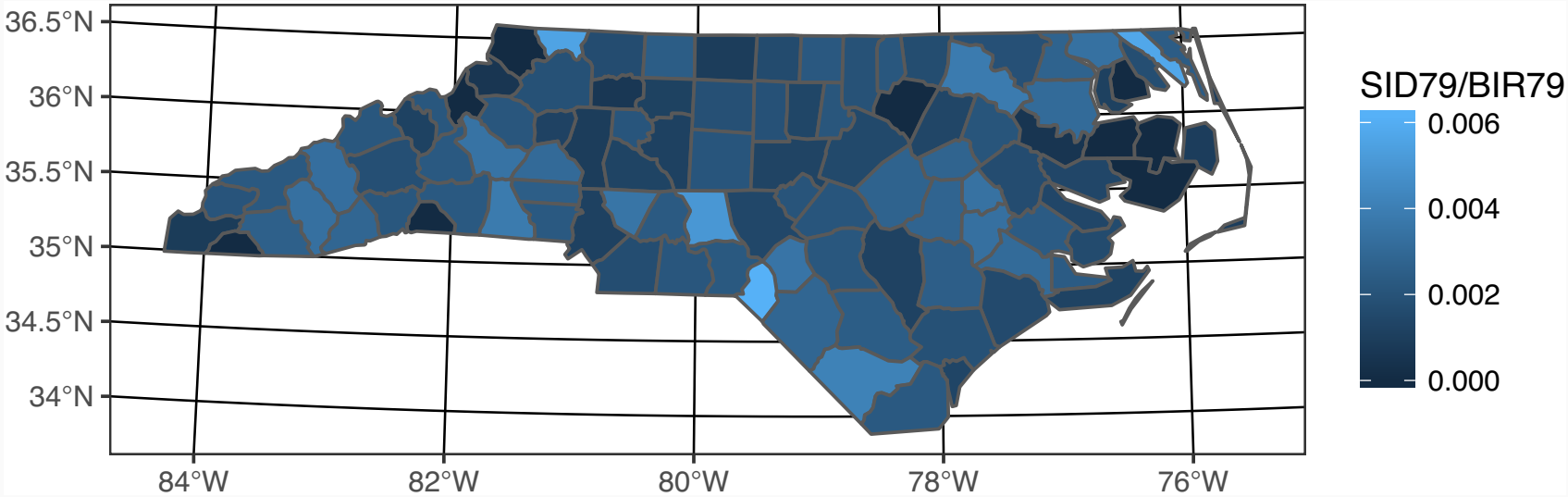
```
devtools::install_github("tidyverse/ggplot2")
```

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



ggplot2 + projections

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

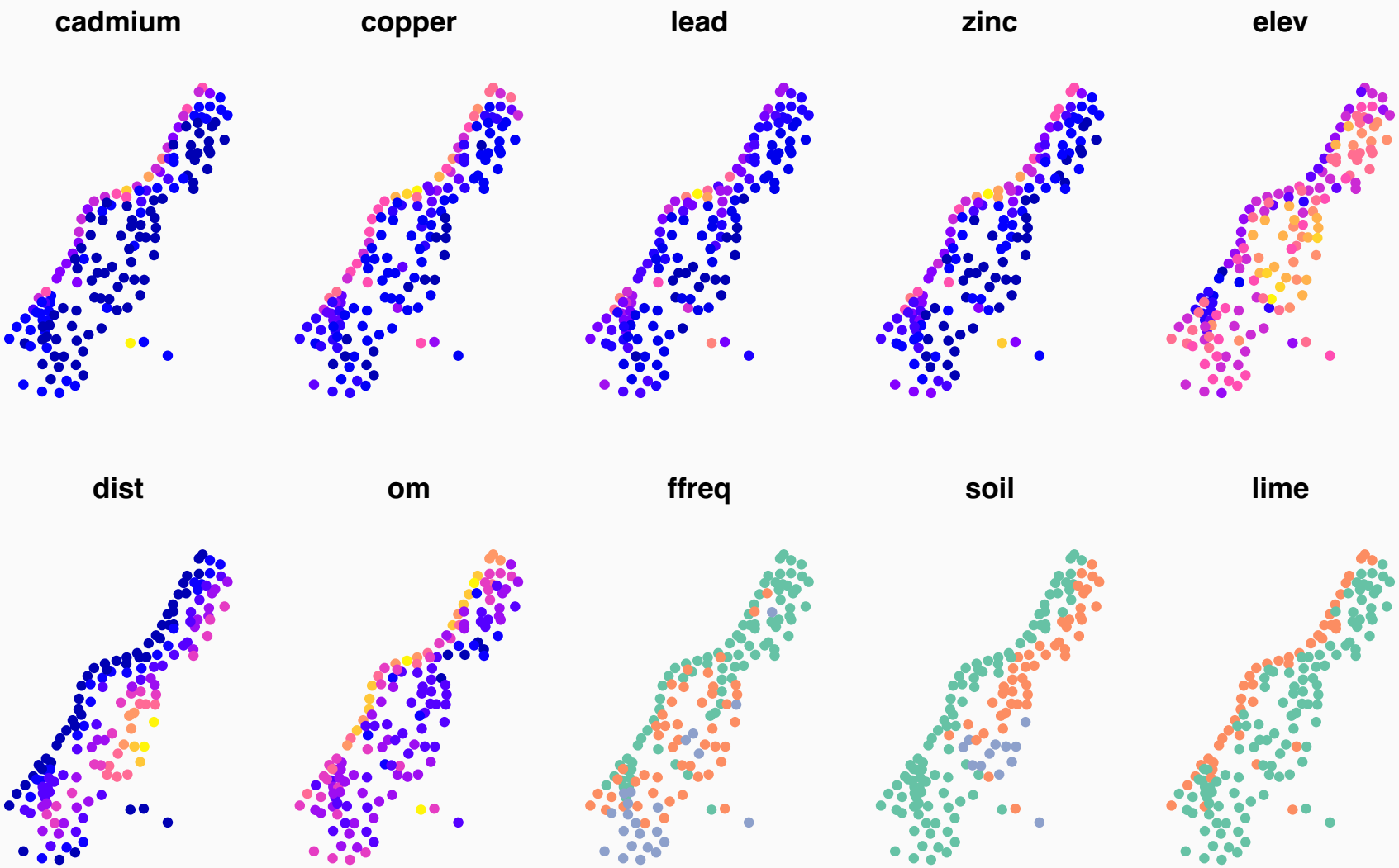


Example Data - Meuse

```
data(meuse, meuse.riv, package="sp")

meuse = st_as_sf(meuse, coords=c("x", "y"), crs=28992)
meuse_riv = st_polygon(list(meuse.riv)) %>% st_sfc() %>% st_set_crs(28992)

tbl_df(meuse)
## # A tibble: 155 x 13
##    cadmium copper  lead  zinc  elev    dist    om ffreq soil  lime
##  *   <dbl>  <dbl> <dbl> <dbl> <dbl>   <dbl> <dbl> <fct> <fct> <fct>
##  1   11.7     85.  299. 1022.  7.91 0.00136 13.6  1     1     1
##  2    8.60    81.  277. 1141.  6.98 0.0122  14.0  1     1     1
##  3    6.50    68.  199.  640.  7.80 0.103   13.0  1     1     1
##  4    2.60    81.  116.  257.  7.66 0.190    8.00 1     2     0
##  5    2.80    48.  117.  269.  7.48 0.277    8.70 1     2     0
##  6    3.00    61.  137.  281.  7.79 0.364    7.80 1     2     0
##  7    3.20    31.  132.  346.  8.22 0.190    9.20 1     2     0
##  8    2.80    29.  150.  406.  8.49 0.0922   9.50 1     1     0
##  9    2.40    37.  133.  347.  8.67 0.185   10.6  1     1     0
## 10    1.60    24.   80.  183.  9.05 0.310    6.30 1     2     0
## # ... with 145 more rows, and 3 more variables: landuse <fct>,
## #   dist.m <dbl>, geometry <sf_geometry [m]>
```

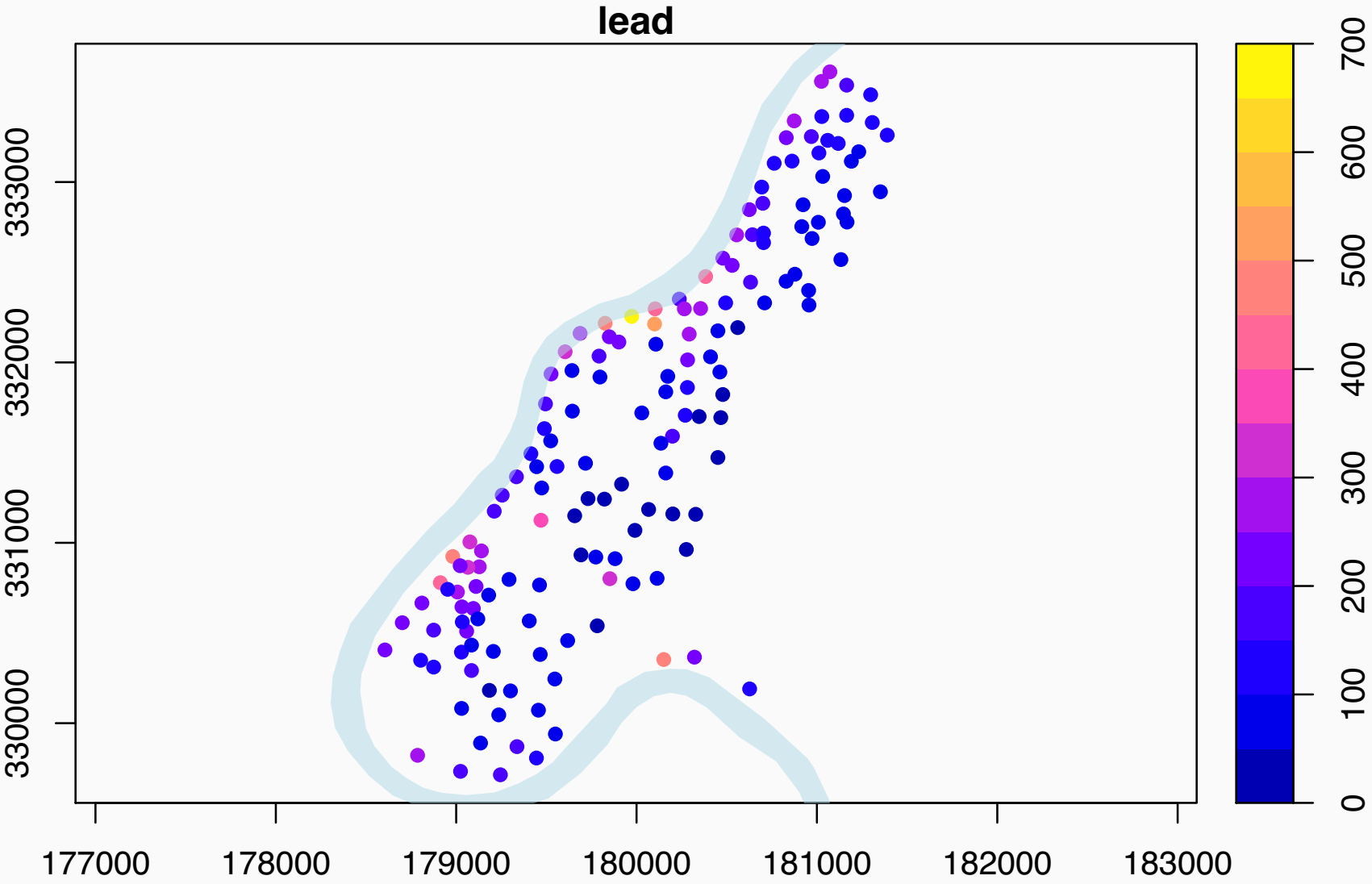
Meuse

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



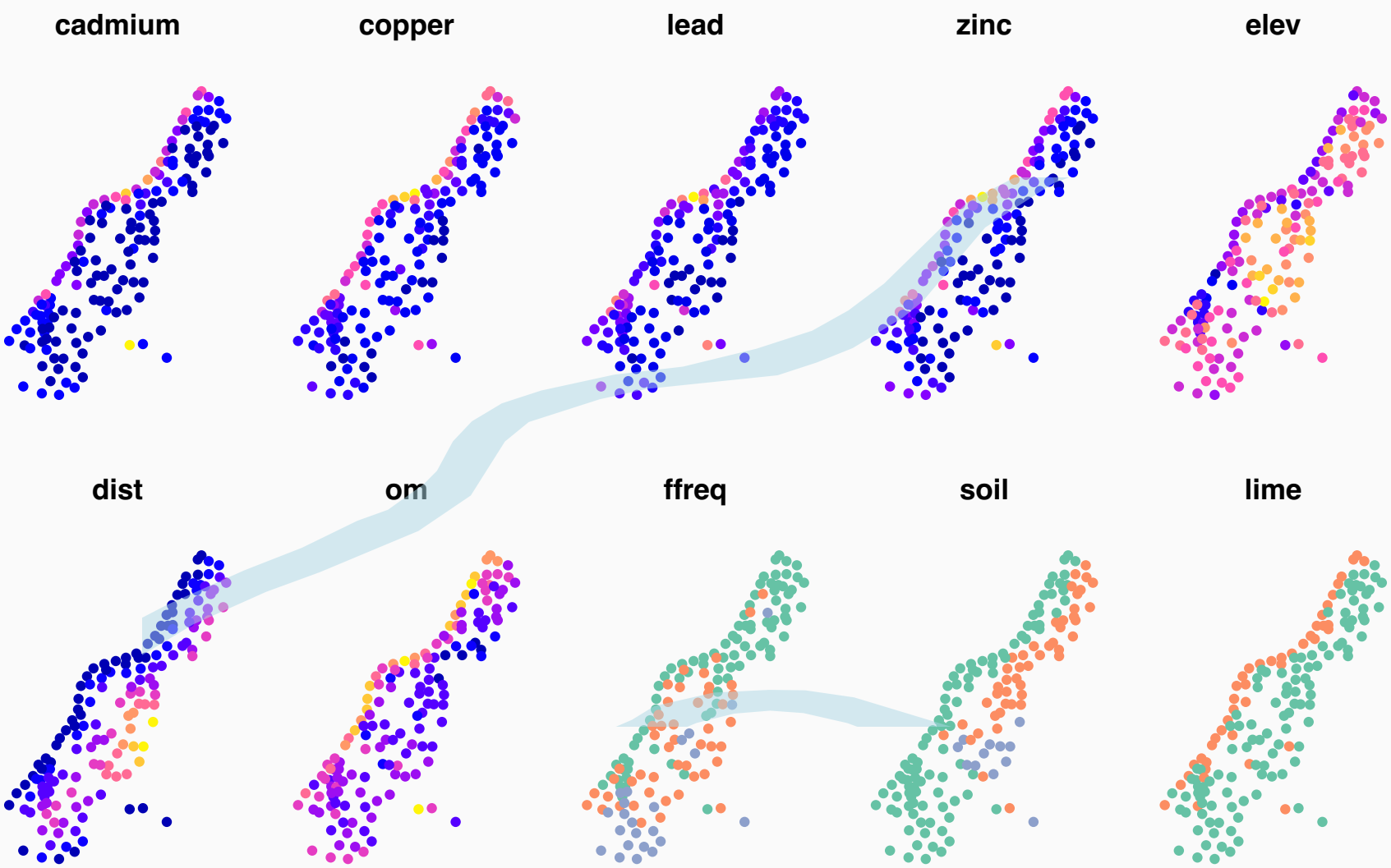
Layering plots

```
plot(meuse[,"lead"], pch=16, axes=TRUE)
plot(meuse_riv, col=adjustcolor("lightblue", alpha.f=0.5), add=TRUE, border = NA)
```



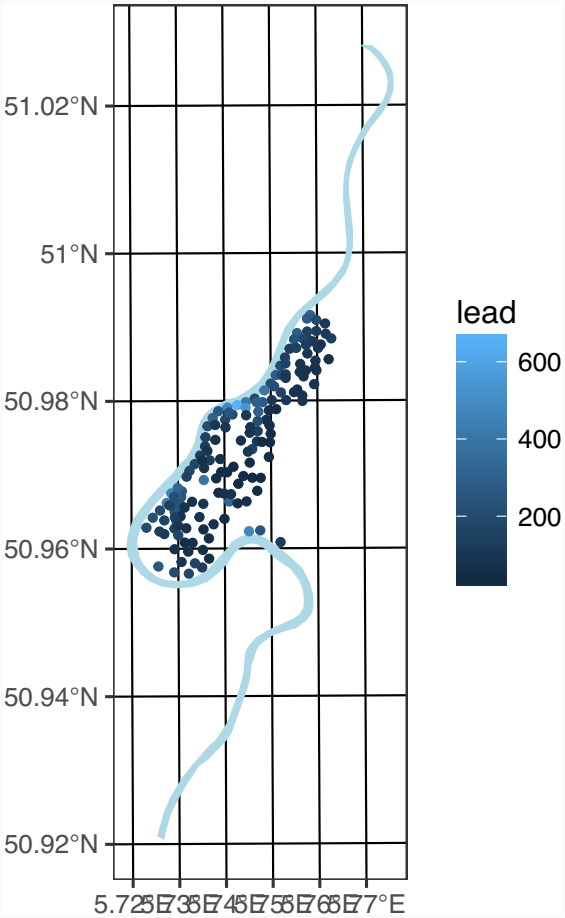
Layering plots (oops)

```
plot(meuse, pch=16)
plot(meuse_riv, col=adjustcolor("lightblue", alpha.f=0.5), add=TRUE, border = NA)
```



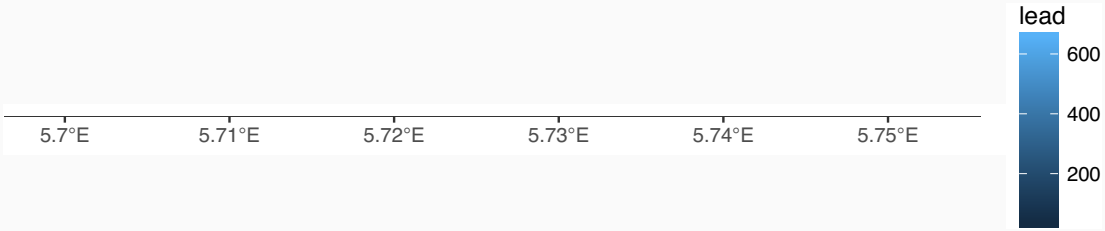
ggplot2

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



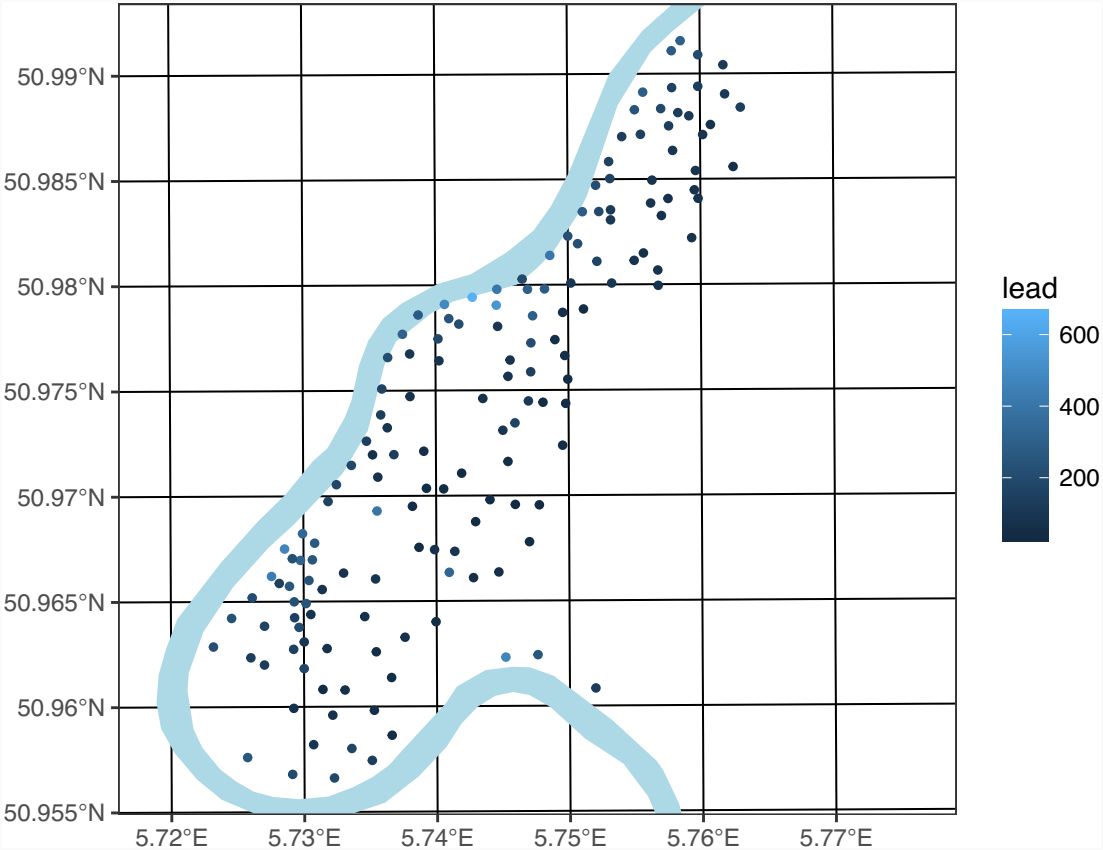
ggplot2 - axis limits

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



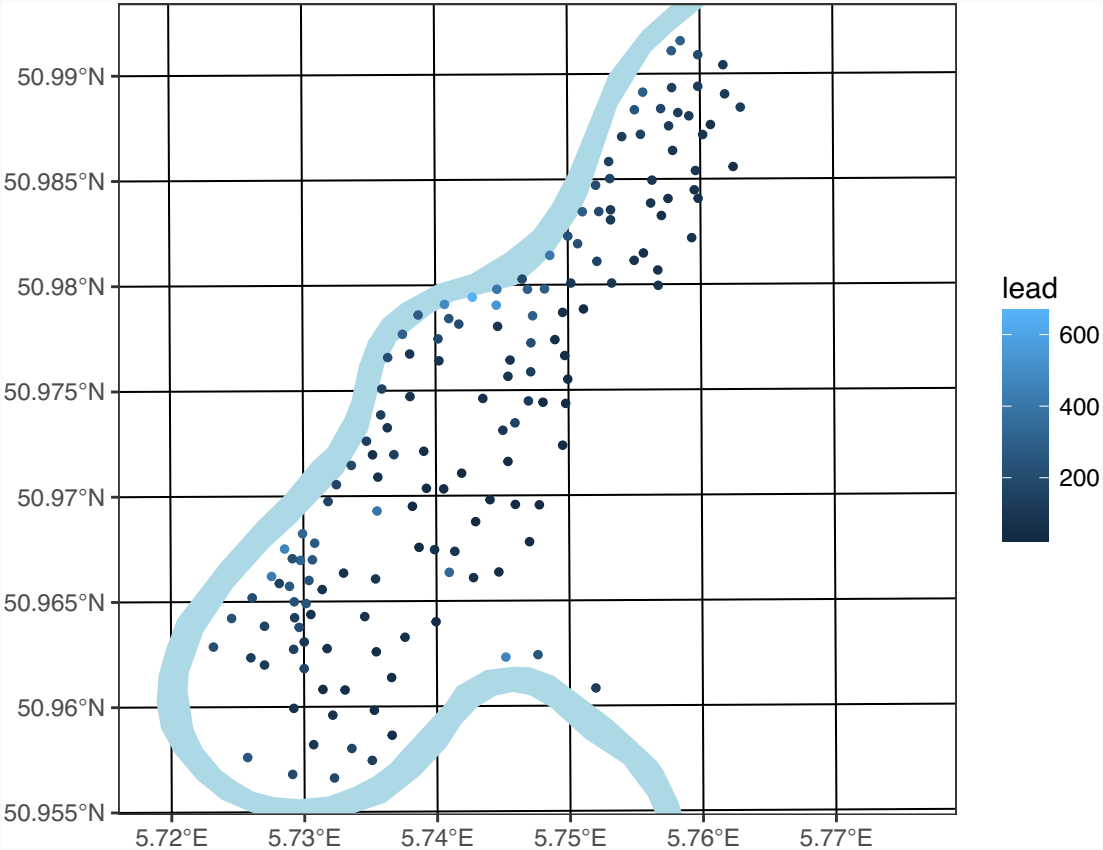
ggplot2 - axis limits

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



ggplot2 - bounding box

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

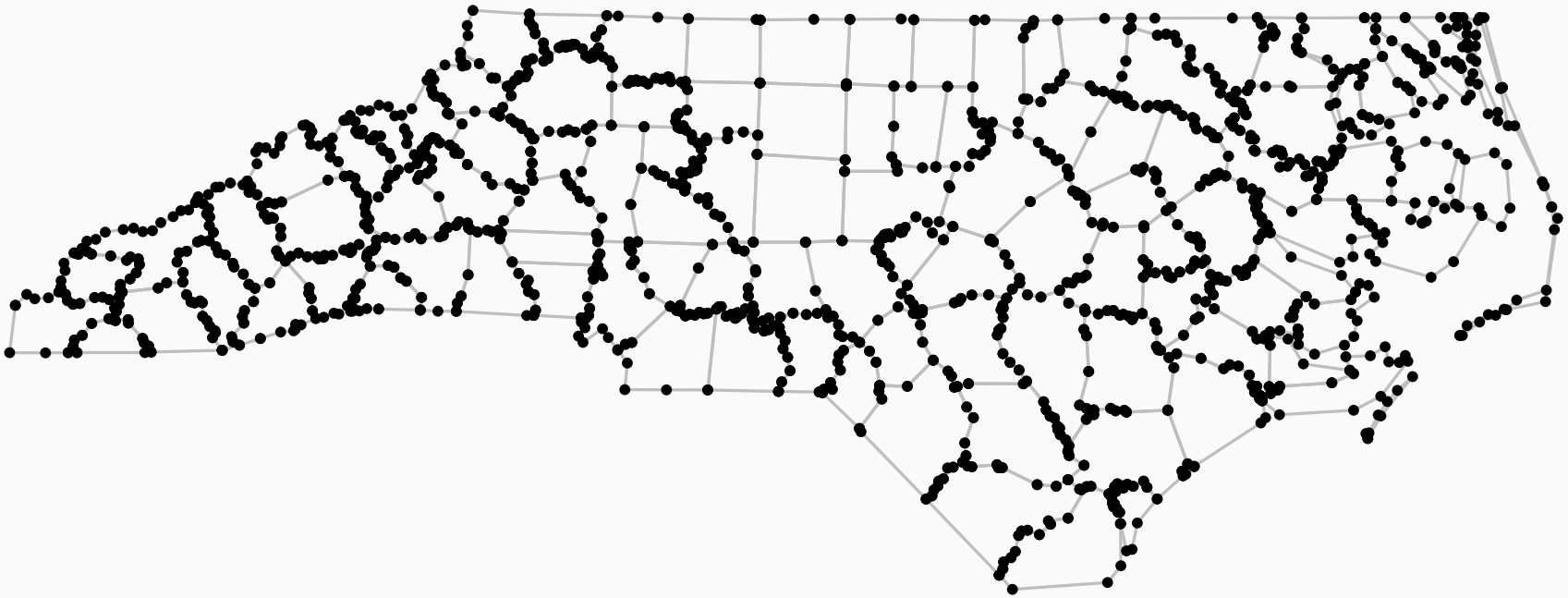


Geometry Manipulation

Casting

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

tbl_df(nc_pts)
## # A tibble: 100 x 8
##    NAME  BIR74 SID74 NWBIR74 BIR79 SID79 NWBIR79                  geometry
##  * <fct> <dbl> <dbl>   <dbl> <dbl> <dbl>   <dbl>    <sf_geometry [degree]>
##  1 Ashe  1091.    1.     10. 1364.    0.     19. MULTIPOINT (-81.47276 36~
##  2 Alle~  487.    0.     10.  542.    3.     12. MULTIPOINT (-81.23989 36~
##  3 Surry 3188.    5.    208. 3616.    6.    260. MULTIPOINT (-80.45634 36~
##  4 Curr~  508.    1.    123.  830.    2.    145. MULTIPOINT (-76.00897 36~
##  5 Nort~ 1421.    9.   1066. 1606.    3.   1197. MULTIPOINT (-77.21767 36~
##  6 Hert~ 1452.    7.    954. 1838.    5.   1237. MULTIPOINT (-76.74506 36~
##  7 Camd~  286.    0.    115.  350.    2.    139. MULTIPOINT (-76.00897 36~
##  8 Gates  420.    0.    254.  594.    2.    371. MULTIPOINT (-76.56251 36~
##  9 Warr~  968.    4.    748. 1190.    2.    844. MULTIPOINT (-78.30876 36~
## 10 Stok~ 1612.    1.    160. 2038.    5.    176. MULTIPOINT (-80.02567 36~
## # ... with 90 more rows
```

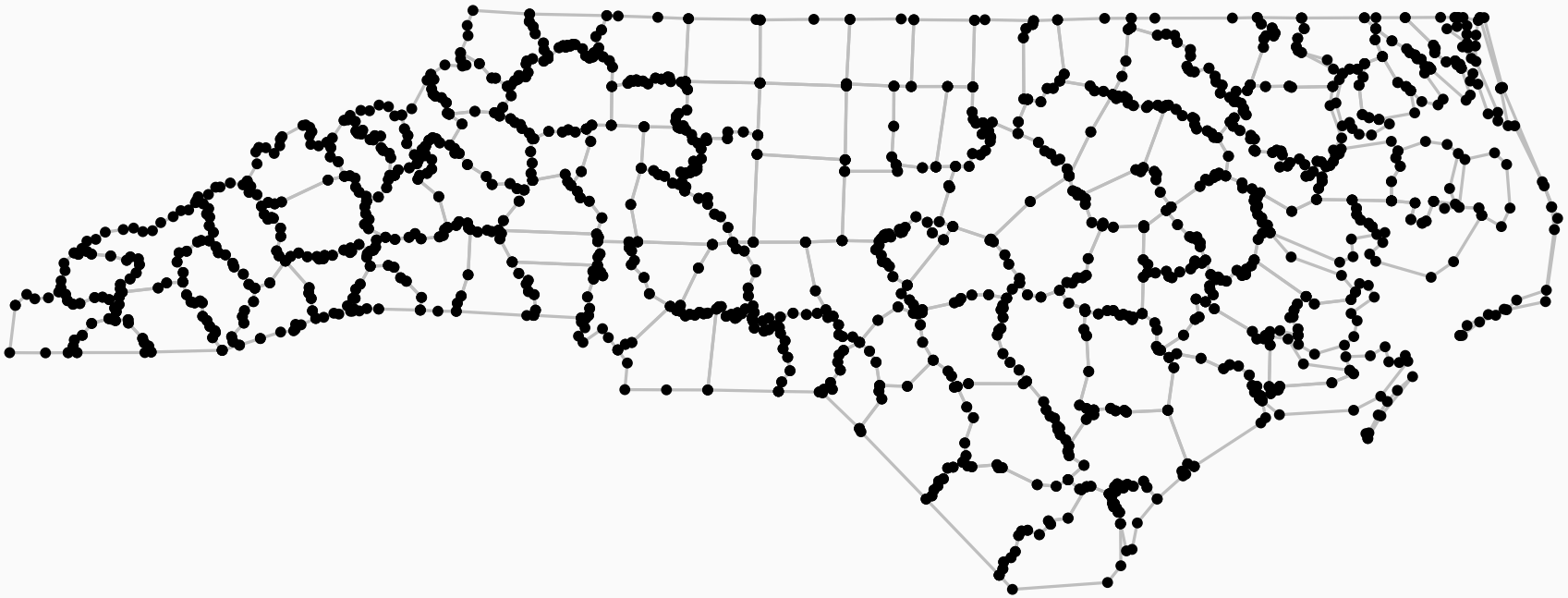
```r
plot(st_geometry(nc), border='grey')
plot(st_geometry(nc_pts), pch=16, cex=0.5, add=TRUE)
```



Casting - POINT

```
st_cast(nc, "POINT")
## Simple feature collection with 2529 features and 7 fields
## geometry type:  POINT
## dimension:      XY
## bbox:           xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax: 36.58965
## epsg (SRID):    4267
## proj4string:    +proj=longlat +datum=NAD27 +no_defs
## First 10 features:
##    NAME BIR74 SID74 NWBIR74 BIR79 SID79 NWBIR79                    geometry
## 1  Ashe  1091     1      10  1364     0      19 POINT (-81.47276 36.23436)
## 2  Ashe  1091     1      10  1364     0      19 POINT (-81.54084 36.27251)
## 3  Ashe  1091     1      10  1364     0      19 POINT (-81.56198 36.27359)
## 4  Ashe  1091     1      10  1364     0      19 POINT (-81.63306 36.34069)
## 5  Ashe  1091     1      10  1364     0      19 POINT (-81.74107 36.39178)
## 6  Ashe  1091     1      10  1364     0      19 POINT (-81.69828 36.47178)
## 7  Ashe  1091     1      10  1364     0      19  POINT (-81.7028 36.51934)
## 8  Ashe  1091     1      10  1364     0      19    POINT (-81.67 36.58965)
## 9  Ashe  1091     1      10  1364     0      19  POINT (-81.3453 36.57286)
## 10 Ashe  1091     1      10  1364     0      19 POINT (-81.34754 36.53791)
```

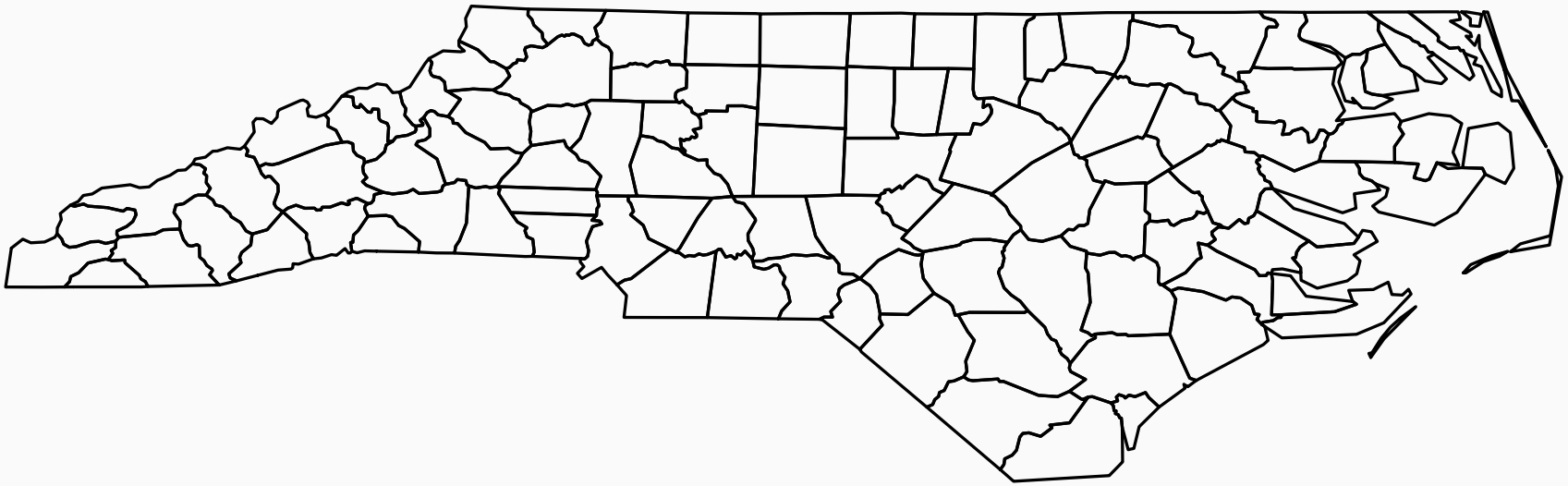
```
plot(st_geometry(nc), border='grey')
plot(st_geometry(st_cast(nc, "POINT")), pch=16, cex=0.5, add=TRUE)
```



Casting - LINESTRING

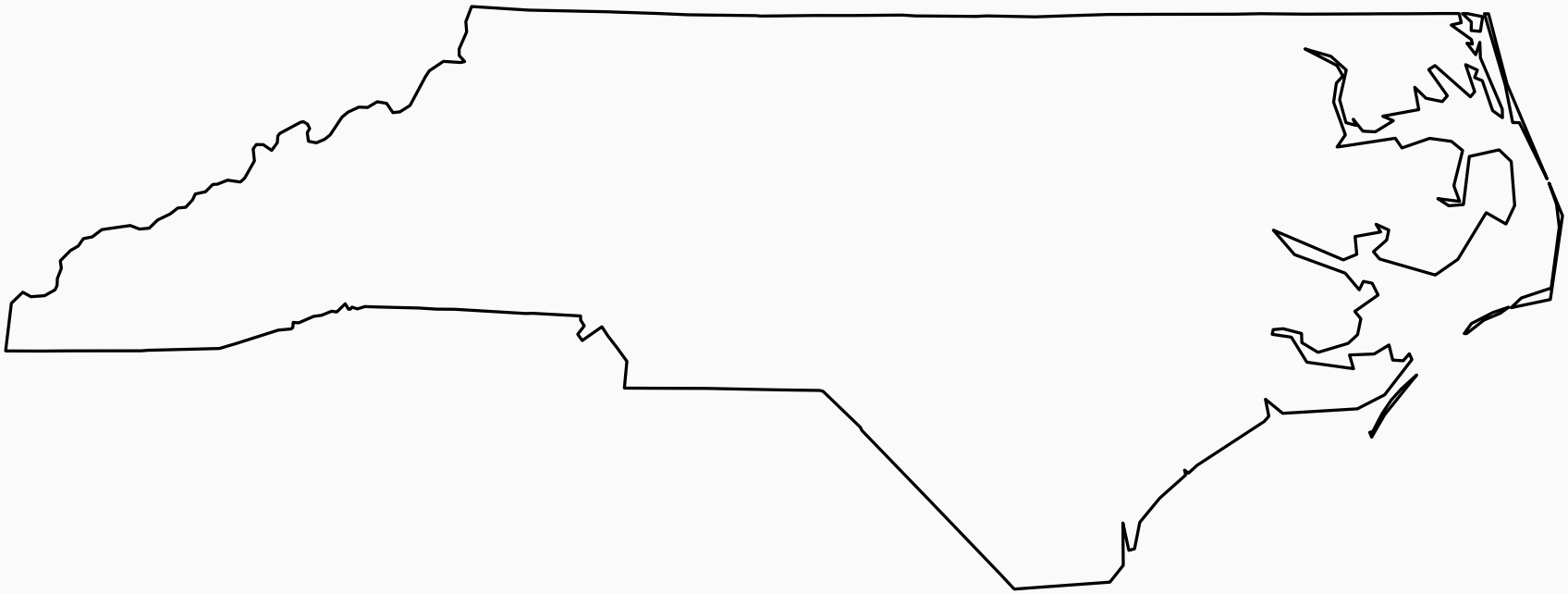
```
st_cast(nc, "MULTILINESTRING") %>% as_tibble()
## # A tibble: 100 x 8
##    NAME  BIR74 SID74 NWBIR74 BIR79 SID79 NWBIR79                  geometry
##  * <fct> <dbl> <dbl>   <dbl> <dbl> <dbl>   <dbl>              <sf_geometry>
##  1 Ashe  1091.    1.     10. 1364.    0.     19. MULTILINESTRING st ((-81~
##  2 Alle~  487.    0.     10.  542.    3.     12. MULTILINESTRING st ((-81~
##  3 Surry 3188.    5.    208. 3616.    6.    260. MULTILINESTRING st ((-80~
##  4 Curr~  508.    1.    123.  830.    2.    145. MULTILINESTRING st ((-76~
##  5 Nort~ 1421.    9.   1066. 1606.    3.   1197. MULTILINESTRING st ((-77~
##  6 Hert~ 1452.    7.    954. 1838.    5.   1237. MULTILINESTRING st ((-76~
##  7 Camd~  286.    0.    115.  350.    2.    139. MULTILINESTRING st ((-76~
##  8 Gates  420.    0.    254.  594.    2.    371. MULTILINESTRING st ((-76~
##  9 Warr~  968.    4.    748. 1190.    2.    844. MULTILINESTRING st ((-78~
## 10 Stok~ 1612.    1.    160. 2038.    5.    176. MULTILINESTRING st ((-80~
## # ... with 90 more rows
```

```
st_cast(nc, "MULTILINESTRING") %>% st_geometry() %>% plot()
```



Grouping Features

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

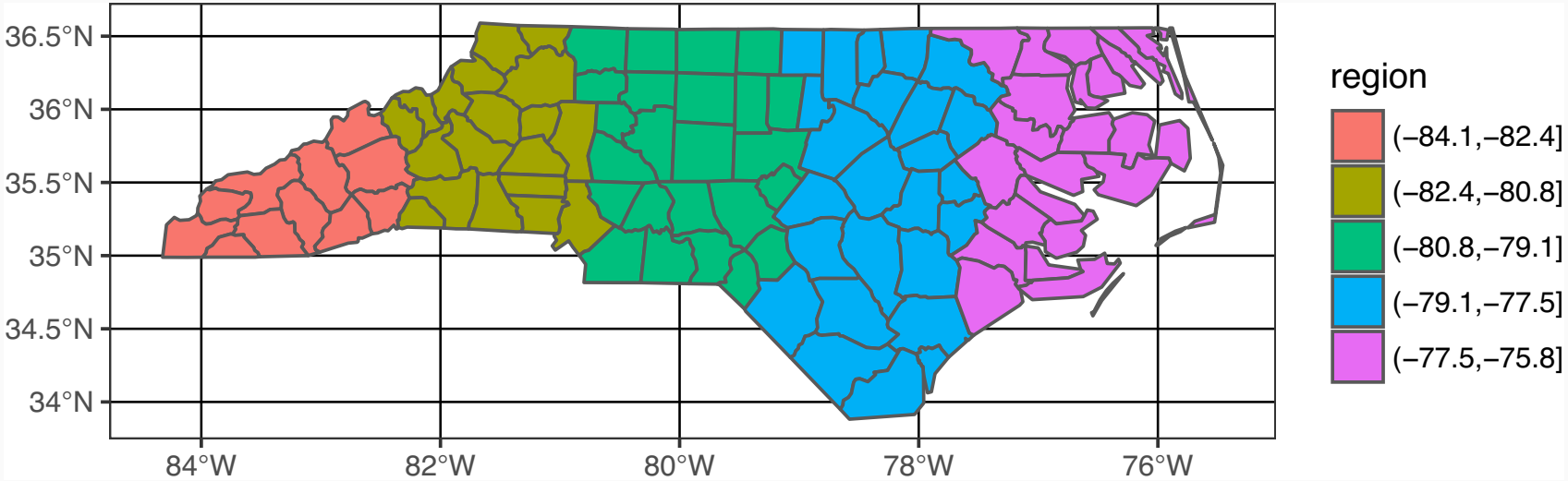


```
nc_state
## Geometry set for 1 feature
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax: 36.58965
## epsg (SRID):    4267
## proj4string:    +proj=longlat +datum=NAD27 +no_defs
## MULTIPOLYGON (((-76.54427 34.58783, -76.55515 3...
```

More Grouping

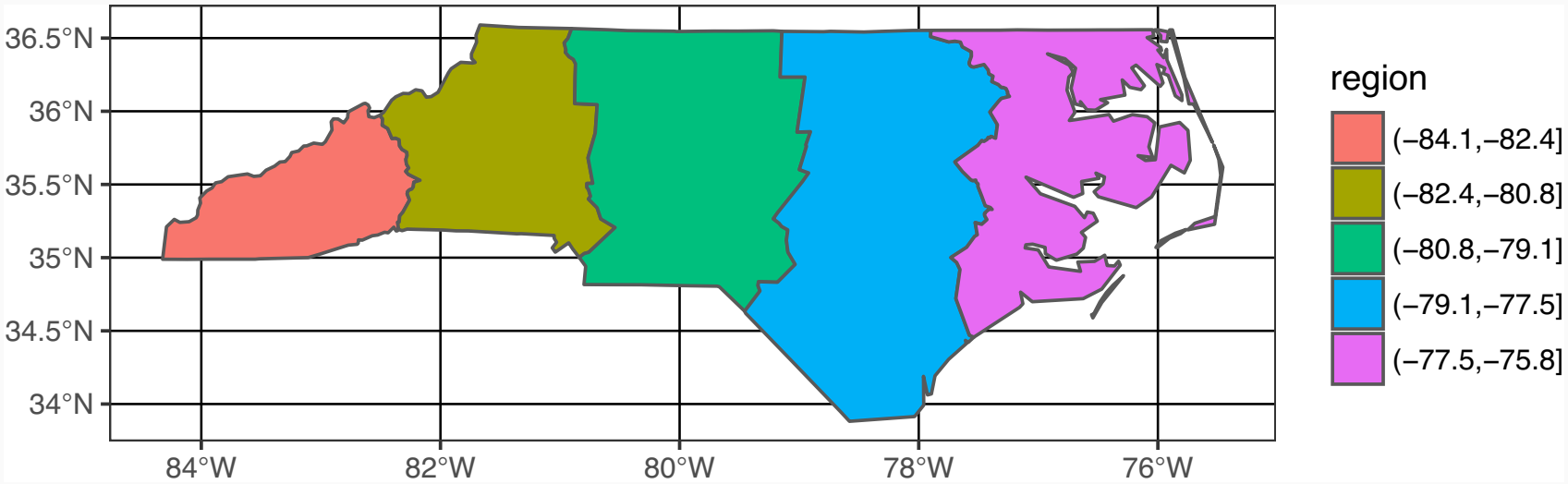
```
nc_cut = nc %>%
  mutate(X = st_centroid(nc) %>% st_coordinates() %>% .[,1]) %>%
  mutate(region = cut(X, breaks = 5))

nc_cut
## Simple feature collection with 100 features and 9 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax: 36.58965
## epsg (SRID):    4267
## proj4string:    +proj=longlat +datum=NAD27 +no_defs
## First 10 features:
##            NAME BIR74 SID74 NWBIR74 BIR79 SID79 NWBIR79         X
## 1          Ashe  1091     1      10  1364     0      19 -81.49826
## 2     Alleghany   487     0      10   542     3      12 -81.12515
## 3         Surry  3188     5     208  3616     6     260 -80.68575
## 4     Currituck   508     1     123   830     2     145 -76.02750
## 5  Northampton  1421     9    1066  1606     3    1197 -77.41056
## 6      Hertford  1452     7     954  1838     5    1237 -76.99478
## 7        Camden   286     0     115   350     2     139 -76.23435
## 8         Gates   420     0     254   594     2     371 -76.70448
## 9        Warren   968     4     748  1190     2     844 -78.11043
## 10       Stokes  1612     1     160  2038     5     176 -80.23428
##            region                       geometry
## 1  (-82.4,-80.8] MULTIPOLYGON (((-81.47276 3...
## 2  (-82.4,-80.8] MULTIPOLYGON (((-81.23989 3...
## 3  (-80.8,-79.1] MULTIPOLYGON (((-80.45634 3...
## 4  (-77.5,-75.8] MULTIPOLYGON (((-76.00897 3...
```

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



dplyr and sf

```
nc_cut %>%
  group_by(region) %>%
  summarize() %>%
  ggplot() +
    geom_sf(aes(fill=region))
```

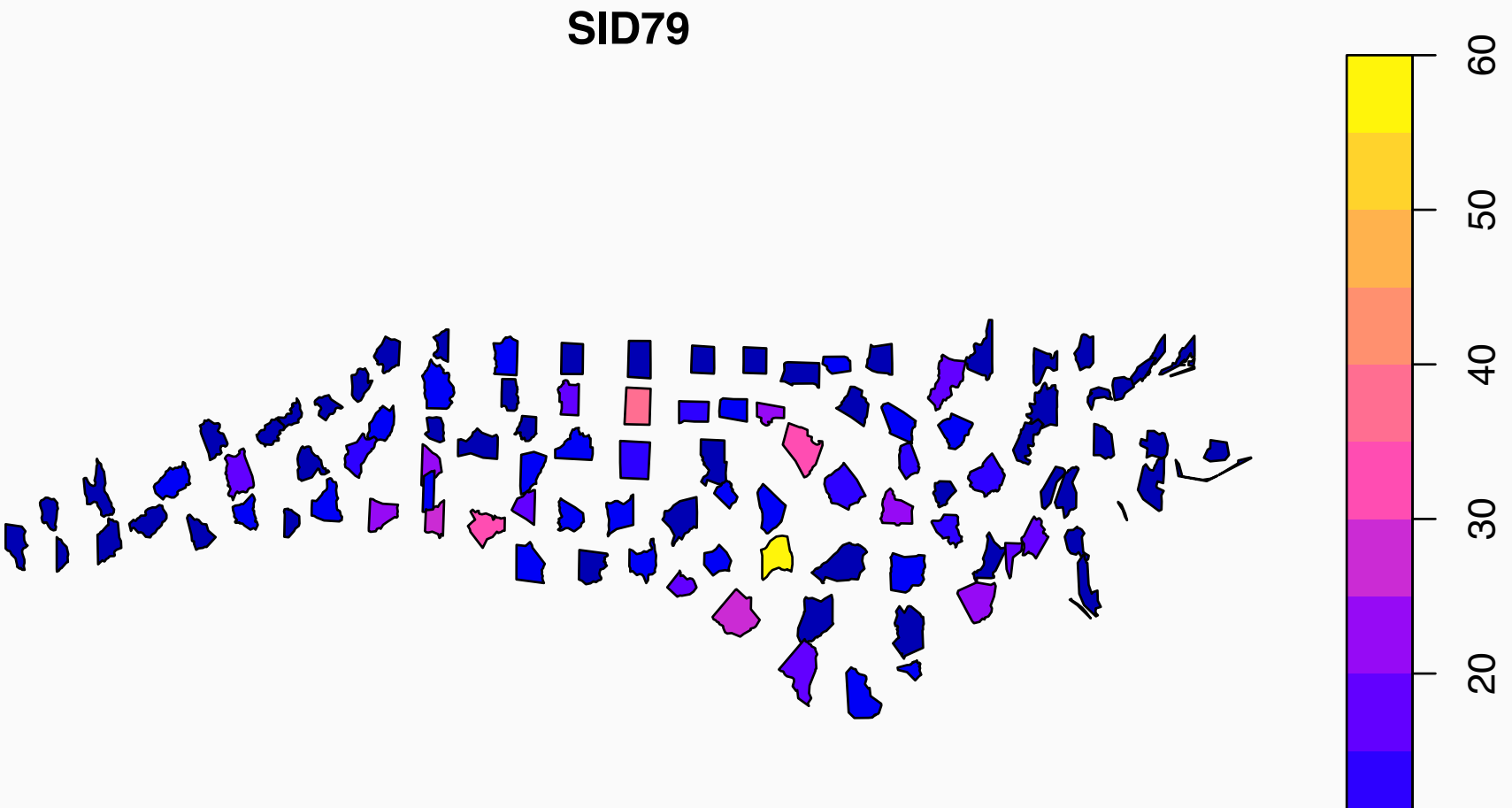


Affine Transfomations

```
rotate = function(a) matrix(c(cos(a), sin(a), -sin(a), cos(a)), 2, 2)

ctrd = st_centroid(nc_state)
state_rotate = (nc_state - ctrd) * rotate(-pi/4) + ctrd
plot(state_rotate, axes=TRUE)
```

Scaling Size

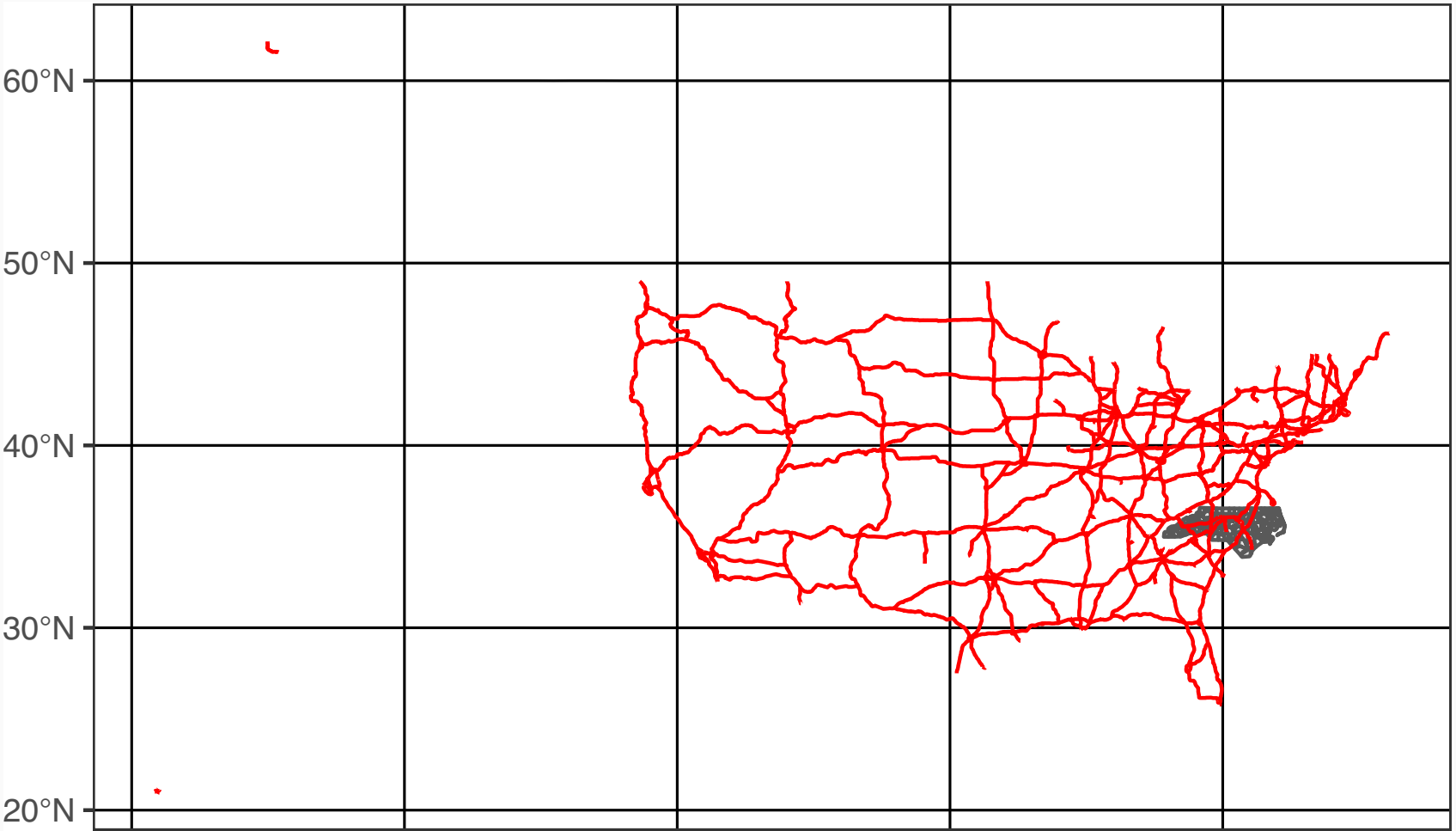
```
ctrd = st_centroid(st_geometry(nc))
area = st_area(nc) %>% strip_attrs()

nc_rot = nc
st_geometry(nc_rot) = (st_geometry(nc) - ctrd) * rotate(pi/2) * .5 + ctrd

plot(nc_rot[,"SID79"])
```



Highway Example

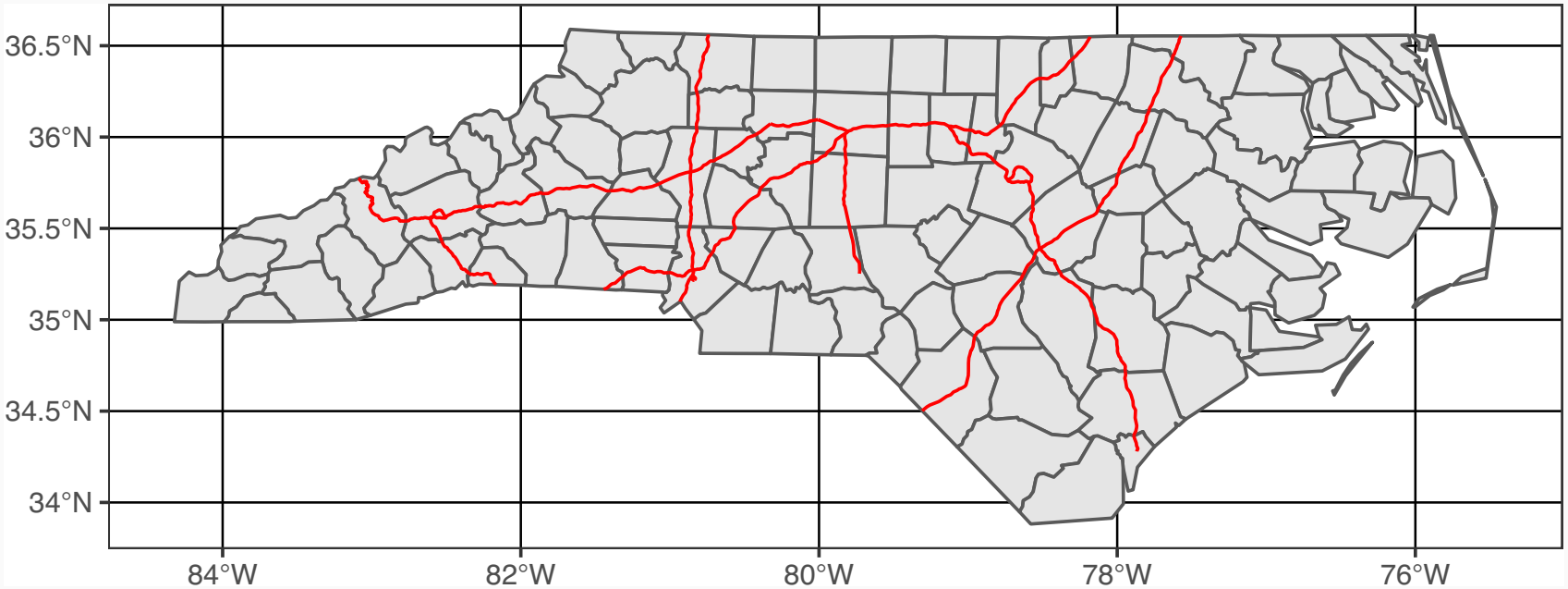
Highways

```
hwy = st_read("../data/gis/us_interstates/", quiet=TRUE, stringsAsFactors=FALSE) %>% s

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



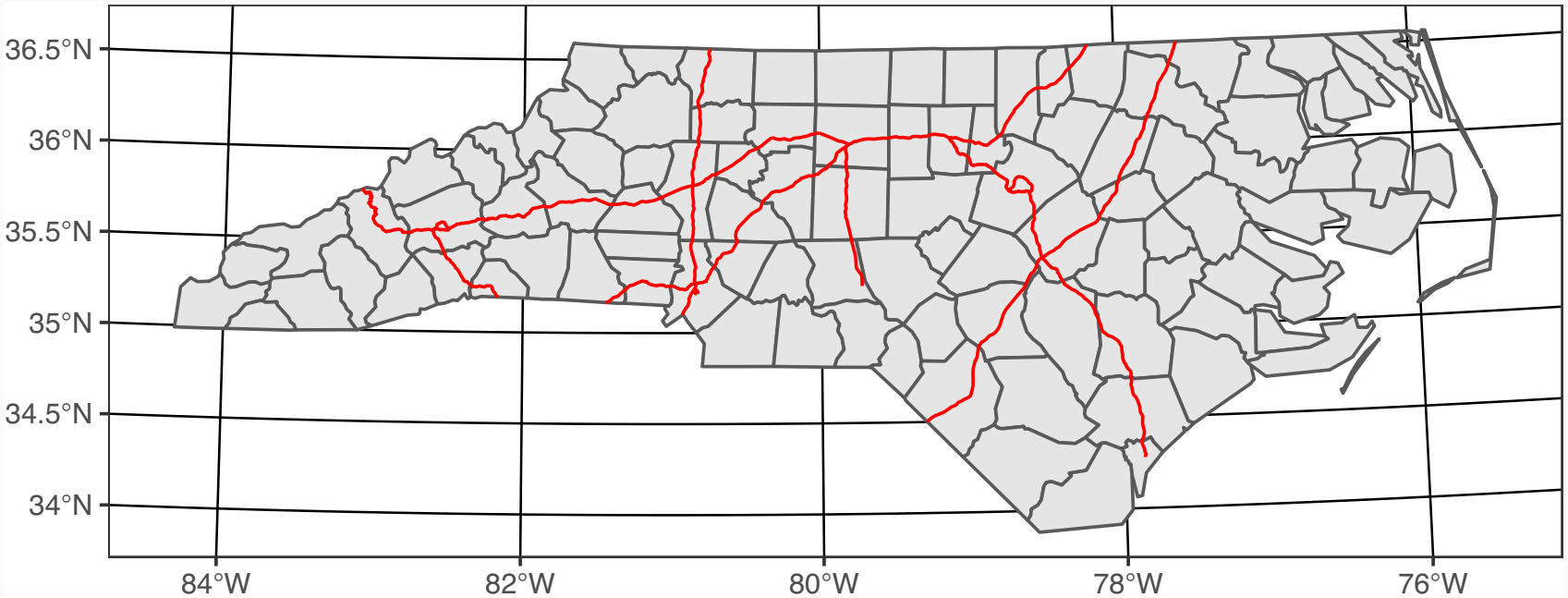
NC Interstate Highways

```
hwy_nc = st_intersection(hwy, nc)
## although coordinates are longitude/latitude, st_intersection assumes that they are

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



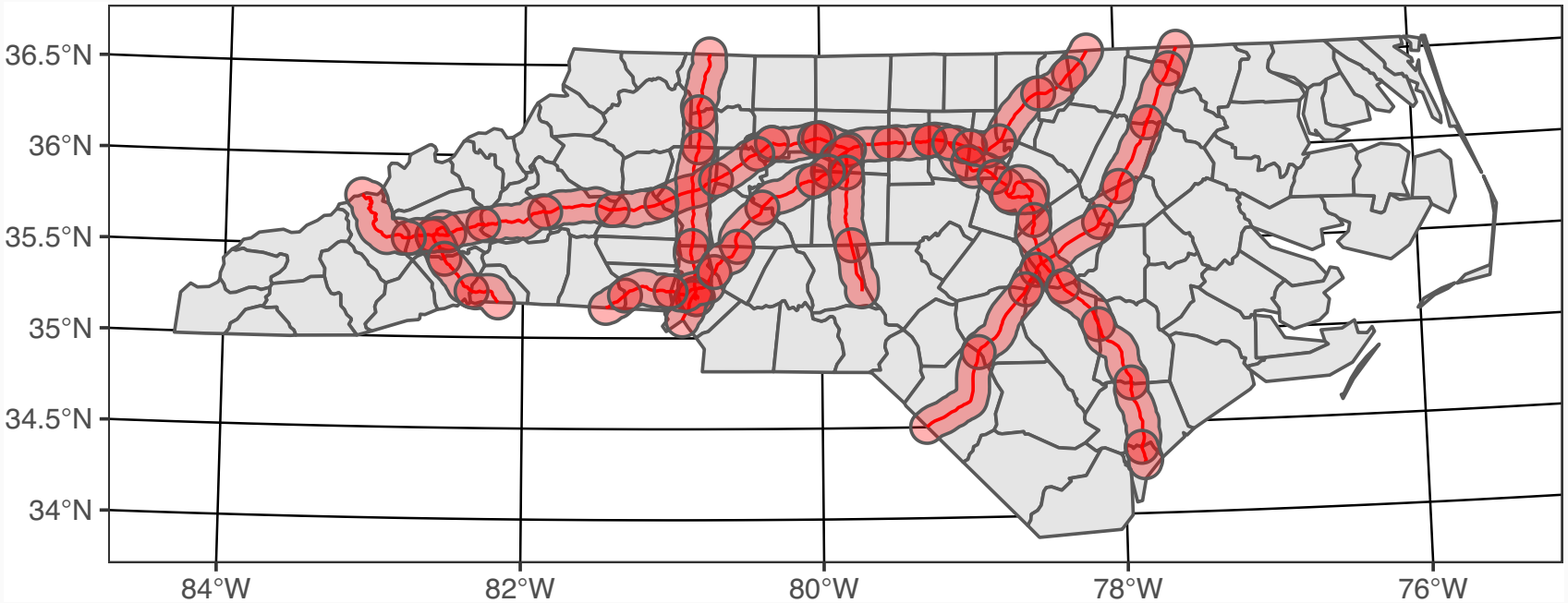
Counties near the interstate (Projection)

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

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



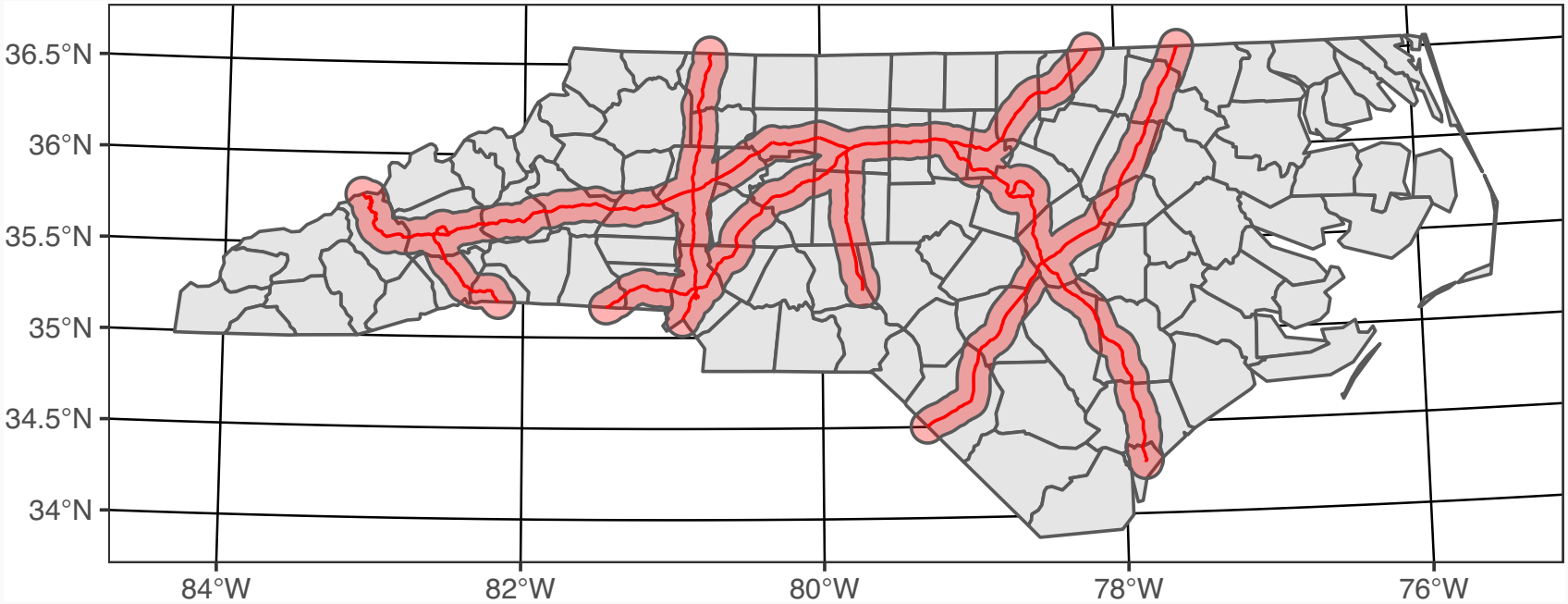
Counties near the interstate (Buffering)

```
hwy_nc_buffer = hwy_nc %>%
  st_transform("+proj=utm +zone=17 +datum=NAD83 +units=m +no_defs") %>%
  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_transform("+proj=utm +zone=17 +datum=NAD83 +units=m +no_defs") %>%
  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)
```



Example

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

Example

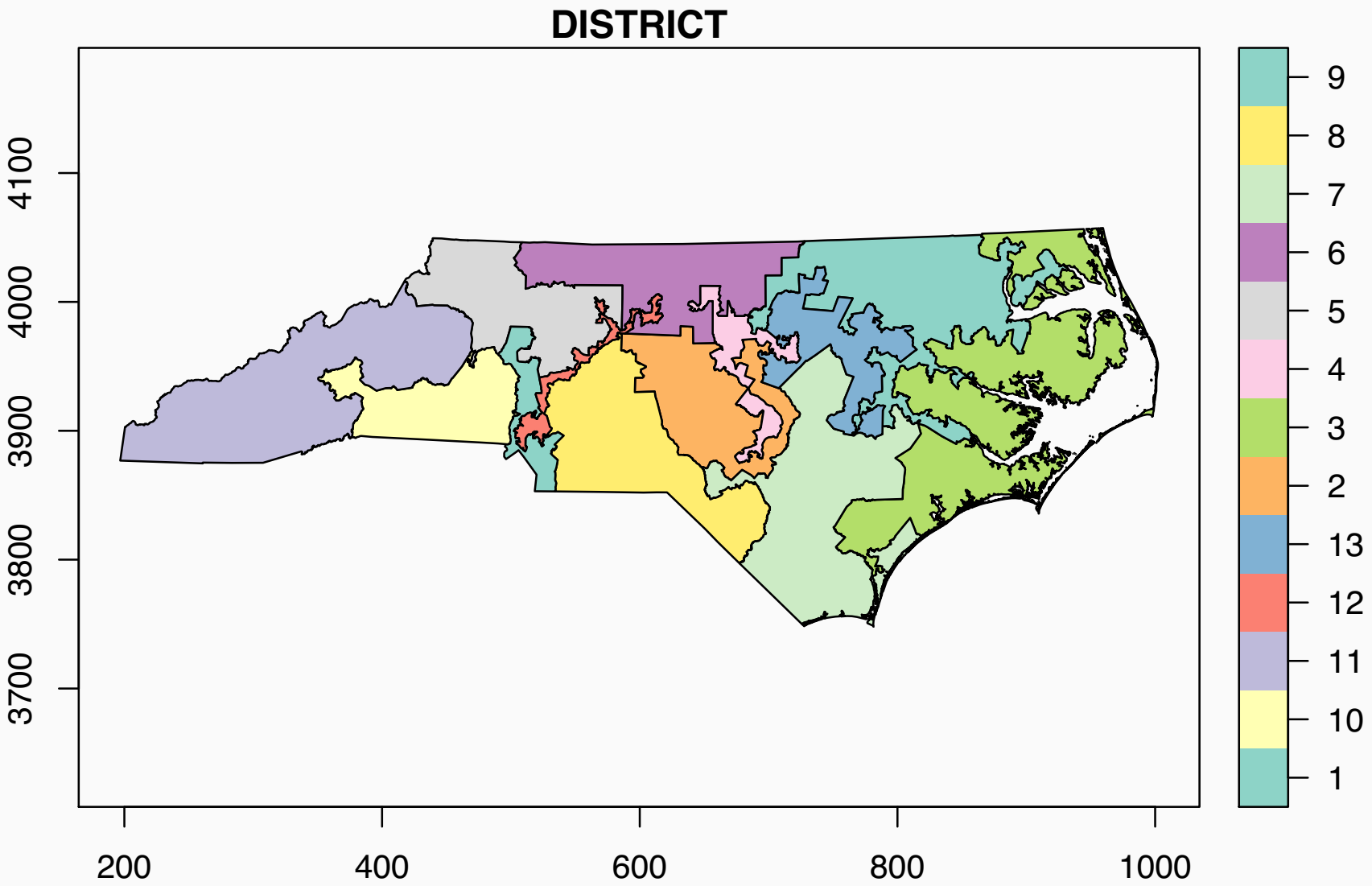
How many counties in North Carolina are within 5, 10, 20, or 50 km of an interstate highway?

Gerrymandering Example

2014 NC House Districts

```
nc_house = st_read("../data/nc_districts114.shp", stringsAsFactors = FALSE,
  select(ID, DISTRICT)

tbl_df(nc_house)
## # A tibble: 13 x 3
##    ID           DISTRICT                        geometry
##    <chr>        <chr>                <sf_geometry [degree]>
##  1 037113114002 2        MULTIPOLYGON (((-80.05325 3...
##  2 037113114003 3        MULTIPOLYGON (((-75.52398 3...
##  3 037113114004 4        MULTIPOLYGON (((-79.47249 3...
##  4 037113114001 1        MULTIPOLYGON (((-76.68697 3...
##  5 037113114005 5        MULTIPOLYGON (((-81.91805 3...
##  6 037113114006 6        MULTIPOLYGON (((-80.97462 3...
##  7 037113114007 7        MULTIPOLYGON (((-79.37719 3...
##  8 037113114008 8        MULTIPOLYGON (((-80.72606 3...
##  9 037113114009 9        MULTIPOLYGON (((-81.10803 3...
## 10 037113114010 10       MULTIPOLYGON (((-82.6516 35...
## 11 037113114011 11       MULTIPOLYGON (((-84.3218 34...
## 12 037113114012 12       MULTIPOLYGON (((-80.97461 3...
## 13 037113114013 13       MULTIPOLYGON (((-78.87711 3...
```

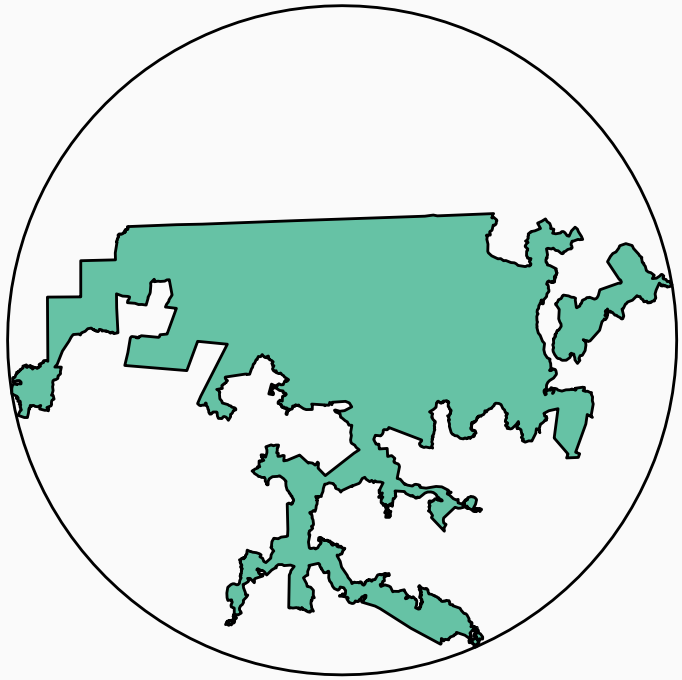
```r
nc_house = nc_house %>%
  st_transform("+proj=utm +zone=17 +datum=NAD83 +units=km +no_defs")
plot(nc_house[,"DISTRICT"], axes=TRUE)
```



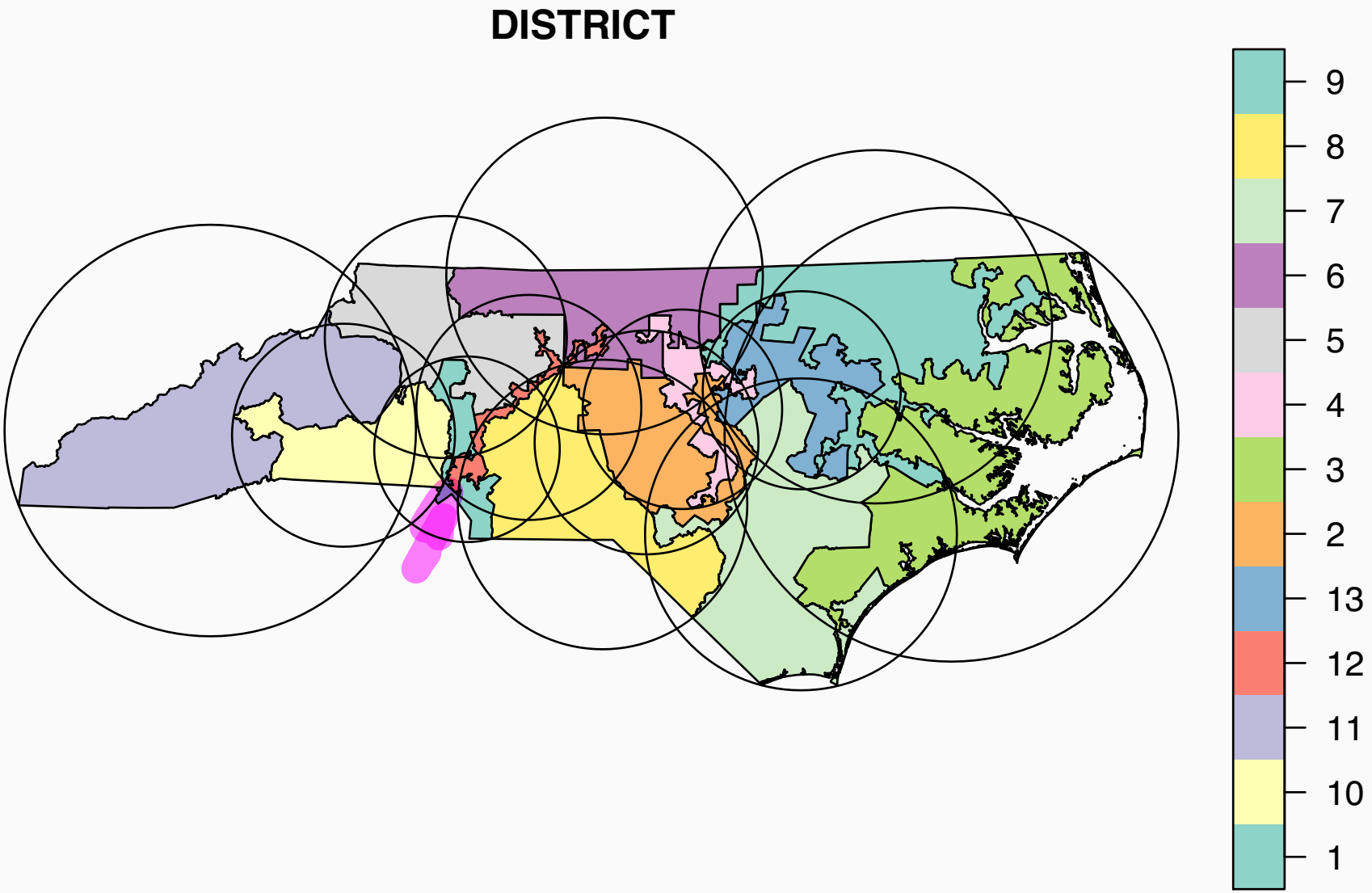
Measuring Compactness - Reock Score

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

```
circs = nc_house %>% st_geometry() %>% lwgeom::st_minimum_bounding_circle()

sub = nc_house$DISTRICT == 1
plot(circs[sub])
plot(nc_house[sub,"DISTRICT"], add=TRUE)
```

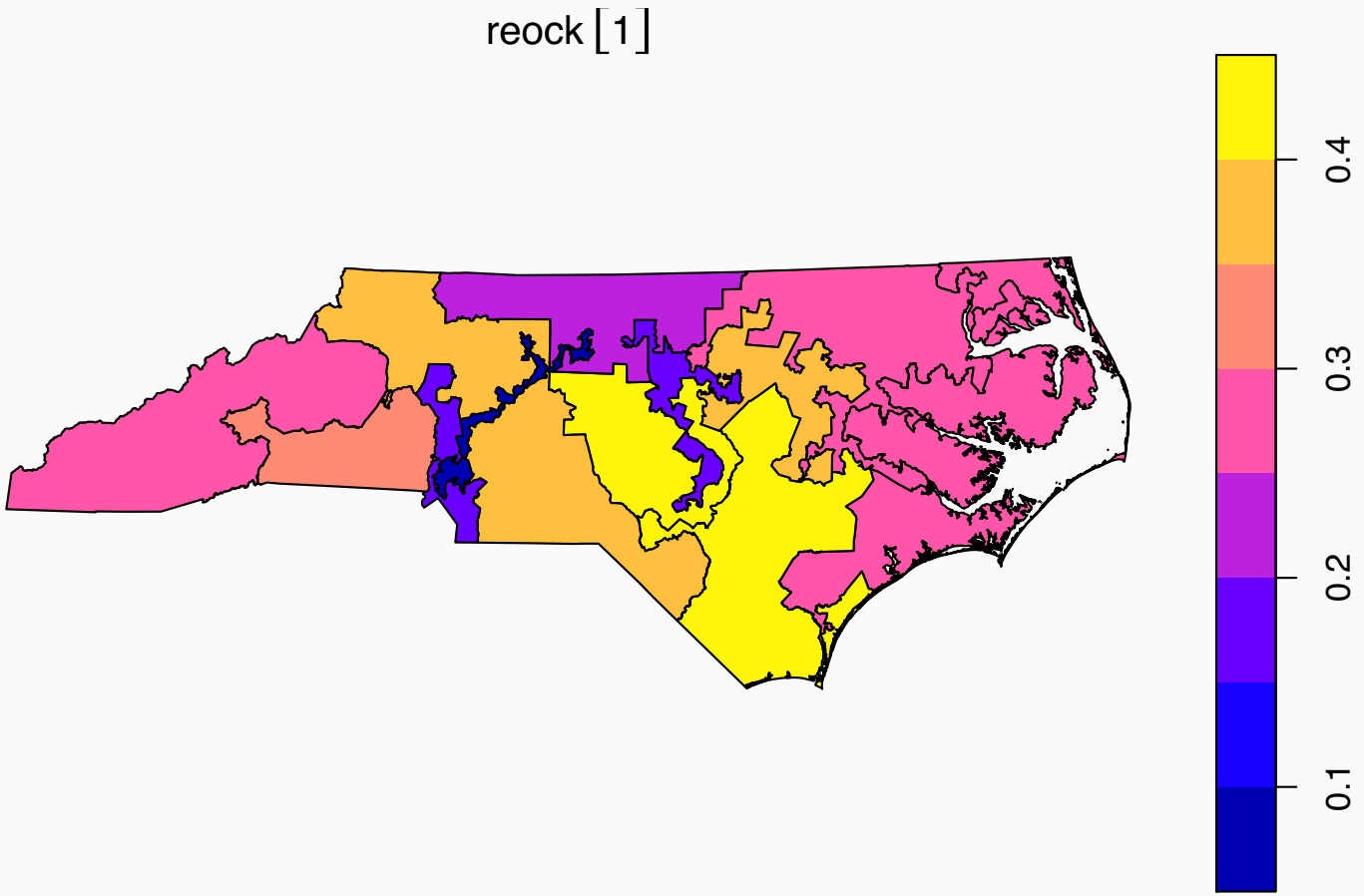


```
plot(nc_house[,"DISTRICT"])
plot(circs,add=TRUE)
```



Calculating Reock

```
nc_house = nc_house %>%
  mutate(reock = st_area(nc_house) / st_area(circs))
plot(nc_house[,"reock"])
```



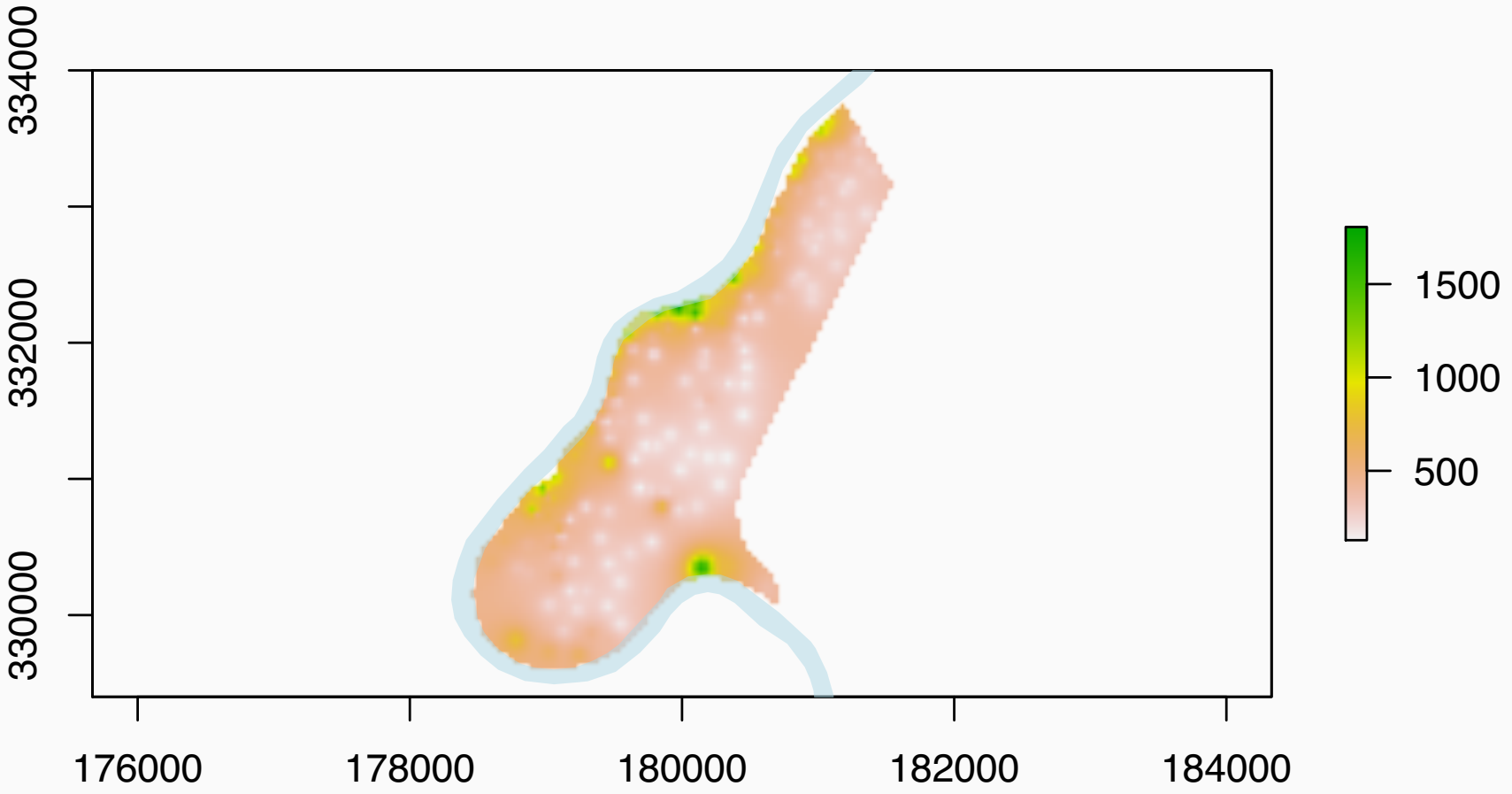
```
tbl_df(nc_house) %>%
  arrange(reock) %>%
  print(n=13)
## # A tibble: 13 x 4
##    ID           DISTRICT reock                                    geometr
##    <chr>        <chr>    <S3: units>                        <sf_geometry [km]
##  1 037113114012 12       0.07119972158781 26 MULTIPOLYGON (((502.31 3899..
##  2 037113114009 9        0.169405525617443  MULTIPOLYGON (((490.2361 39..
##  3 037113114004 4        0.1735809490213    MULTIPOLYGON (((637.4776 39..
##  4 037113114006 6        0.240919191926239  MULTIPOLYGON (((502.2744 40..
##  5 037113114003 3        0.251285019523225  MULTIPOLYGON (((995.1797 39..
##  6 037113114011 11       0.264255107593438  MULTIPOLYGON (((196.7812 38..
##  7 037113114001 1        0.289934134507595  MULTIPOLYGON (((888.2895 40..
##  8 037113114010 10       0.34012606752961   MULTIPOLYGON (((350.3919 39..
##  9 037113114008 8        0.353232490504049  MULTIPOLYGON (((524.9335 38..
## 10 037113114013 13       0.382195549931454  MULTIPOLYGON (((691.9403 39..
## 11 037113114005 5        0.397082589710882  MULTIPOLYGON (((417.5582 40..
## 12 037113114007 7        0.414888641986656  MULTIPOLYGON (((648.136 387..
## 13 037113114002 2        0.42590009492903   MULTIPOLYGON (((585.5425 39..
```

Raster Data

Example data - Meuse

```
meuse_rast = raster(system.file("external/test.grd", package="raster"))

meuse_rast
## class       : RasterLayer
## dimensions  : 115, 80, 9200  (nrow, ncol, ncell)
## resolution  : 40, 40  (x, y)
## extent      : 178400, 181600, 329400, 334000  (xmin, xmax, ymin, ymax)
## coord. ref. : +init=epsg:28992 +towgs84=565.237,50.0087,465.658,-0.406857,0.350733,
## data source : /usr/local/lib/R/3.4/site-library/raster/external/test.grd
## names       : test
## values      : 128.434, 1805.78  (min, max)
```

```
plot(meuse_rast)
plot(meuse_riv, add=TRUE, col=adjustcolor("lightblue",alpha.f = 0.5), border=NA)
```



raster class

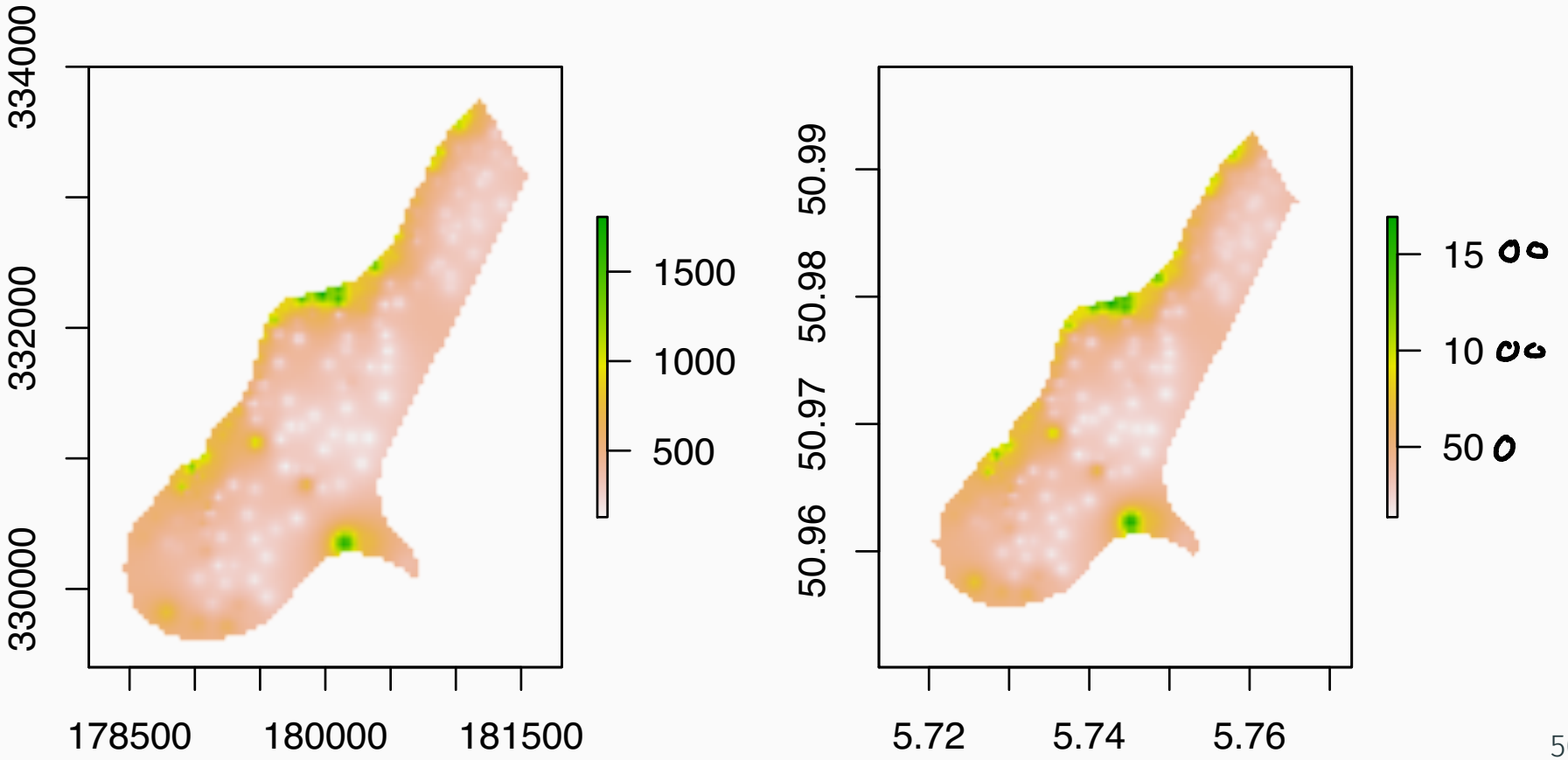
```
str(meuse_rast)
## Formal class 'RasterLayer' [package "raster"] with 12 slots
##   ..@ file    :Formal class '.RasterFile' [package "raster"] with 13 slots
##   .. .. ..@ name        : chr "/usr/local/lib/R/3.4/site-library/raster/external/te
##   .. .. ..@ datanotation: chr "FLT4S"
##   .. .. ..@ byteorder   : Named chr "little"
##   .. .. .. ..- attr(*, "names")= chr "value"
##   .. .. ..@ nodatavalue : num -3.4e+38
##   .. .. ..@ NAchanged   : logi FALSE
##   .. .. ..@ nbands      : int 1
##   .. .. ..@ bandorder   : Named chr "BIL"
##   .. .. .. ..- attr(*, "names")= chr "value"
##   .. .. ..@ offset      : int 0
##   .. .. ..@ toptobottom : logi TRUE
##   .. .. ..@ blockrows   : int 0
##   .. .. ..@ blockcols   : int 0
##   .. .. ..@ driver      : chr "raster"
##   .. .. ..@ open        : logi FALSE
##   ..@ data    :Formal class '.SingleLayerData' [package "raster"] with 13 slots
##   .. .. ..@ values    : logi(0)
##   .. .. ..@ offset    : num 0
##   .. .. ..@ gain      : num 1
##   .. .. ..@ inmemory  : logi FALSE
##   .. .. ..@ fromdisk  : logi TRUE
##   .. .. ..@ isfactor  : logi FALSE
##   .. .. ..@ attributes: list()
##   .. .. ..@ haveminmax: logi TRUE
##   .. .. ..@ min       : num 128
```

raster features

```
extent(meuse_rast)
## class       : Extent
## xmin        : 178400
## xmax        : 181600
## ymin        : 329400
## ymax        : 334000

dim(meuse_rast)
## [1] 115  80   1

res(meuse_rast)
## [1] 40 40

projection(meuse_rast)
## [1] "+init=epsg:28992 +towgs84=565.237,50.0087,465.658,-0.406857,0.350733,-1.87035,

meuse_rast[20,]
##  [1]      NA      NA      NA      NA      NA      NA      NA      NA
##  [9]      NA      NA      NA      NA      NA      NA      NA      NA
## [17]      NA      NA      NA      NA      NA      NA      NA      NA
## [25]      NA      NA      NA      NA      NA      NA      NA      NA
## [33]      NA      NA      NA      NA      NA      NA      NA      NA
## [41]      NA      NA      NA      NA      NA      NA      NA      NA
## [49]      NA      NA      NA      NA      NA      NA      NA      NA
## [57]      NA      NA      NA 749.536 895.292 791.145 607.186 511.044
## [65] 468.404 399.325 350.362 306.180 300.483 310.082 283.940 285.771
## [73] 304.709 309.690 301.799 308.753 328.357 345.611      NA      NA
```

Rasters and Projections

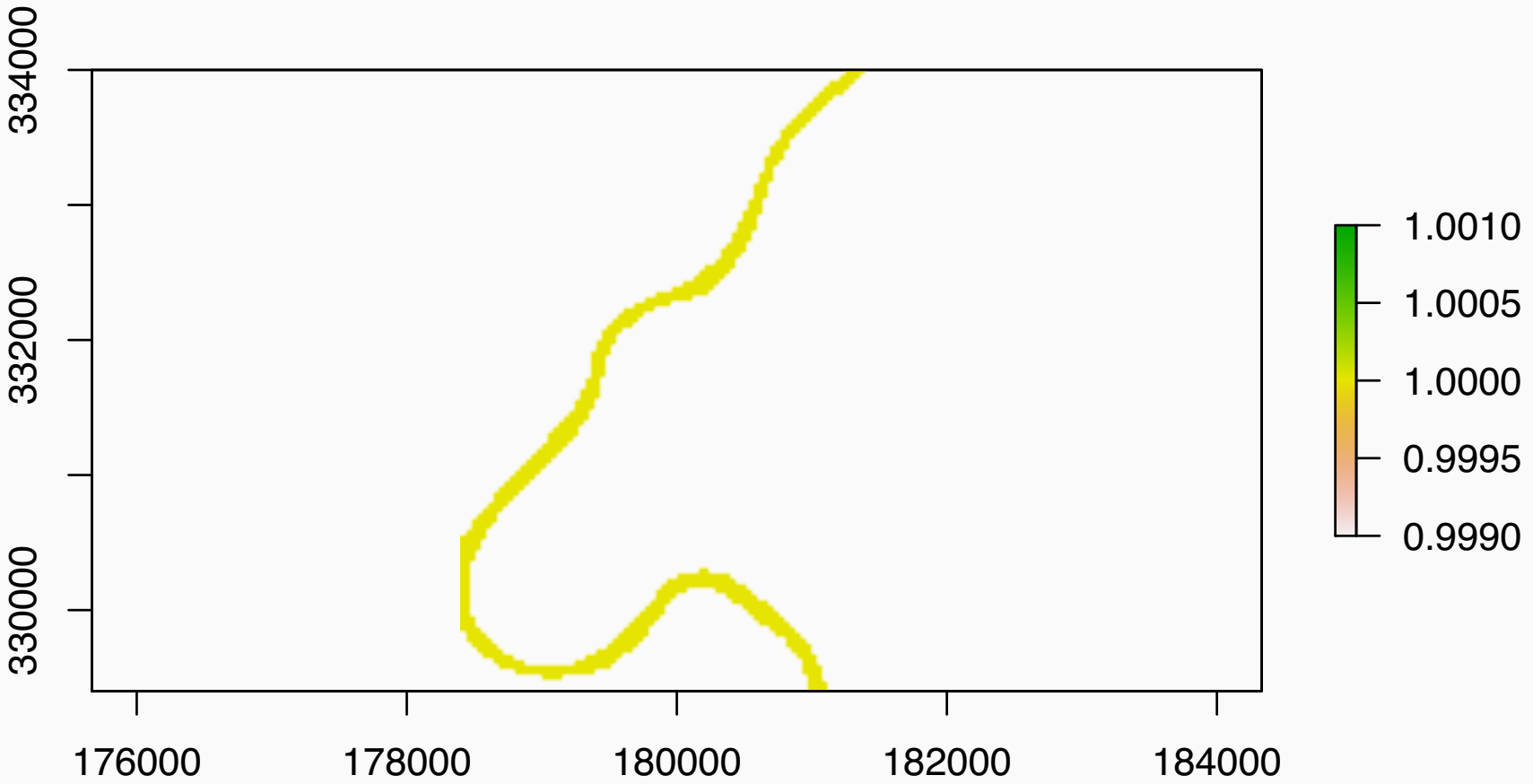
```
library(rgdal)
meuse_rast_ll = projectRaster(meuse_rast, crs="+proj=longlat +datum=NAD27 +no_defs")

par(mfrow=c(1,2))
plot(meuse_rast)
plot(meuse_rast_ll)
```



```
meuse_rast
## class       : RasterLayer
## dimensions  : 115, 80, 9200  (nrow, ncol, ncell)
## resolution  : 40, 40  (x, y)
## extent      : 178400, 181600, 329400, 334000  (xmin, xmax, ymin, ymax)
## coord. ref. : +init=epsg:28992 +towgs84=565.237,50.0087,465.658,-0.406857
## data source : /usr/local/lib/R/3.4/site-library/raster/external/test.grd
## names       : test
## values      : 128.434, 1805.78  (min, max)

meuse_rast_ll
## class       : RasterLayer
## dimensions  : 131, 91, 11921  (nrow, ncol, ncell)
## resolution  : 0.000569, 0.00036  (x, y)
## extent      : 5.717362, 5.769141, 50.95089, 50.99805  (xmin, xmax, ymin,
## coord. ref. : +proj=longlat +datum=NAD27 +no_defs +ellps=clrk66 +nadgrids
## data source : in memory
## names       : test
## values      : 135.647, 1693.578  (min, max)
```

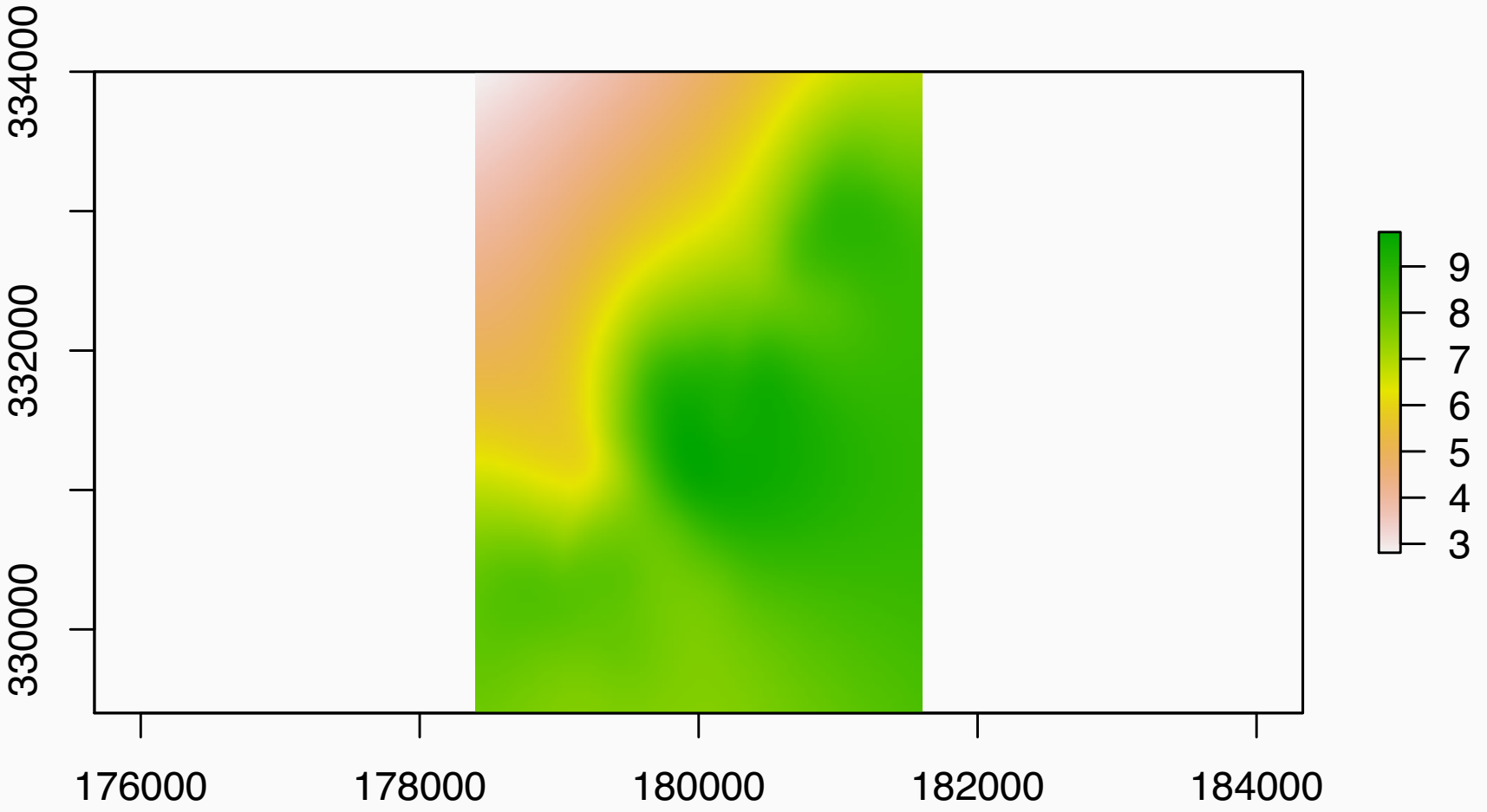
Simple Features ⟷ Rasters

```
meuse_riv_rast = rasterize(meuse_riv, meuse_rast)
## Error in (function (classes, fdef, mtable) : unable to find an inherited method for

meuse_riv_rast = rasterize(as(meuse_riv, "Spatial"), meuse_rast)
plot(meuse_riv_rast)
```



```
head(meuse)
## Simple feature collection with 6 features and 12 fields
## geometry type:  POINT
## dimension:      XY
## bbox:           xmin: 181025 ymin: 333260 xmax: 181390 ymax: 333611
## epsg (SRID):    28992
## proj4string:    +proj=sterea +lat_0=52.15616055555555 +lon_0=5.38763888888889 +k=0.
##   cadmium copper lead zinc  elev       dist   om ffreq soil lime landuse
## 1    11.7     85  299 1022 7.909 0.00135803 13.6     1    1    1      Ah
## 2     8.6     81  277 1141 6.983 0.01222430 14.0     1    1    1      Ah
## 3     6.5     68  199  640 7.800 0.10302900 13.0     1    1    1      Ah
## 4     2.6     81  116  257 7.655 0.19009400  8.0     1    2    0      Ga
## 5     2.8     48  117  269 7.480 0.27709000  8.7     1    2    0      Ah
## 6     3.0     61  137  281 7.791 0.36406700  7.8     1    2    0      Ga
##   dist.m              geometry
## 1     50 POINT (181072 333611)
## 2     30 POINT (181025 333558)
## 3    150 POINT (181165 333537)
## 4    270 POINT (181298 333484)
## 5    380 POINT (181307 333330)
## 6    470 POINT (181390 333260)

head(st_coordinates(meuse))
##        X      Y
## 1 181072 333611
## 2 181025 333558
## 3 181165 333537
## 4 181298 333484
```

```r
library(fields)

tps = Tps(x = st_coordinates(meuse), Y=meuse$elev)
pred_grid = xyFromCell(meuse_rast, seq_along(meuse_rast))

meuse_elev_pred = meuse_rast
meuse_elev_pred[] = predict(tps, pred_grid)

plot(meuse_elev_pred)
```



ggplot and rasters

```
p = rasterToPolygons(meuse_elev_pred) %>% st_as_sf()

(ggplot() + geom_sf(data=meuse, aes(color=elev), size=1)) +
(ggplot() + geom_sf(data=p, aes(fill=test), color=NA))
```