

```
library(sf)
library(spdep)
library(tmap)

NC <- read_sf("data/NC_REGION.shp")

NC_UTM <- st_transform(NC, crs = st_crs("EPSG:26917"))
```

### Question 1:

Use the `poly2nb()` function to create a Rook's case neighbor construct by changing `queen = TRUE` to `queen = FALSE`. Call this new object `rook_nb` in your code. Plot a map using `tmap` that shows the counties, the connections as lines, and the county centroids as points. Include a title (hint: use `main.title` if you want to have the title outside the map frame).

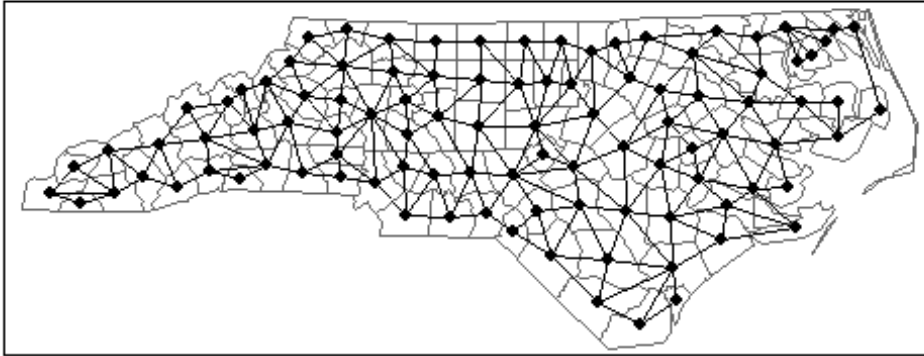
```
rook_nb <- poly2nb(NC_UTM, queen = FALSE)

NC_centroids <- st_centroid(NC_UTM)

## Warning in st_centroid.sf(NC_UTM): st_centroid assumes attributes are
## constant over geometries of x

nb_lines <- nb2lines(
  nb = rook_nb,
  coords = NC_centroids$geometry
)
tm_shape(NC_UTM) +
  tm_borders() +
  tm_shape(nb_lines) + tm_lines() +
  tm_shape(NC_centroids) + tm_dots(size = 0.2) +
  tm_layout(
    main.title = "Rook's Case Neighbors"
  )
```

## Rook's Case Neighbors



### Question 2:

Compare the distribution of links between the Queen's case and Rook's case. What changed when we restricted the neighbor criterion by using a Rook's case? Using the code above, what are **names** of the 'least connected regions' generated with a Rook's case neighborhood? Provide the code used to get your results.

```
queen_names <- NC_UTM %>%  
  tibble::column_to_rownames("NAME") %>%  
  st_as_sf() %>%  
  poly2nb(  
    pl = .,  
    queen = TRUE  
  )
```

```
summary(queen_names)
```

```
## Neighbour list object:  
## Number of regions: 100  
## Number of nonzero links: 490  
## Percentage nonzero weights: 4.9  
## Average number of links: 4.9  
## Link number distribution:  
##  
##  2  3  4  5  6  7  8  9  
##  8 15 17 23 19 14  2  2
```

```
## 8 least connected regions:
## Currituck Chowan Tyrrell Dare Polk Pamlico Clay New Hanover with 2 links
## 2 most connected regions:
## Iredell Moore with 9 links

rook_names <- NC_UTM %>%
  tibble::column_to_rownames("NAME") %>%
  st_as_sf() %>%
  poly2nb(queen = FALSE)

summary(rook_names)

## Neighbour list object:
## Number of regions: 100
## Number of nonzero links: 462
## Percentage nonzero weights: 4.62
## Average number of links: 4.62
## Link number distribution:
##
##  2  3  4  5  6  7  8  9
##  8 18 20 25 21  4  3  1
## 8 least connected regions:
## Currituck Chowan Tyrrell Dare Polk Pamlico Clay New Hanover with 2 links
## 1 most connected region:
## Iredell with 9 links
```

### Question 3:

In addition to  $k = 1$ , use `knn2nb()` to create neighbor constructs for  $k = 2$ ,  $k = 4$ , and  $k = 6$ . Plot the connections for each using `tmap`, including the label (e.g., “ $k = 1$ ”). Provide your R code and plots for the neighbor constructs. Use `tmap_arrange()` to place all 4 plots in a single “image”.

*“Extra credit” if students can do this in a for loop, using `lapply`, `purrr::map`, etc., i.e., not doing it “one at a time”*

```
k_1 <- knn2nb(
  knn = knearneigh(
    x = NC_centroids,
    k = 1
  ),
  row.names = NC_UTM$NAME
) %>% nb2lines(coords = NC_centroids$geometry)

k_2 <- knn2nb(
  knn = knearneigh(
    x = NC_centroids,
    k = 2
  ),
  row.names = NC_UTM$NAME
```

```

) %>% nb2lines(coords = NC_centroids$geometry)

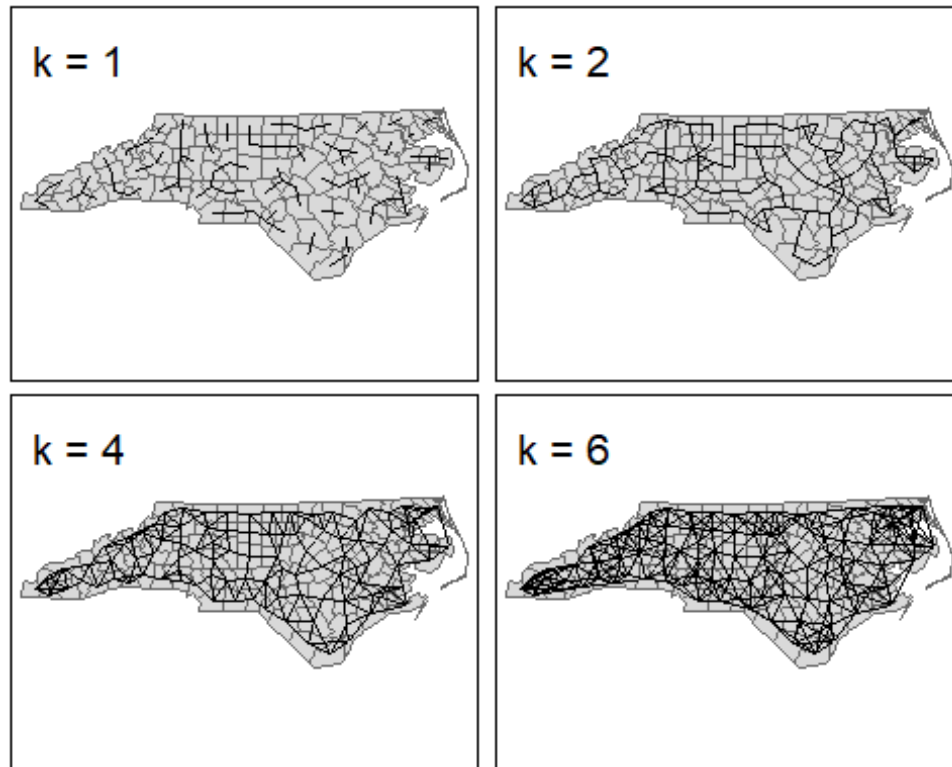
k_4 <- knn2nb(
  knn = knearneigh(
    x = NC_centroids,
    k = 4
  ),
  row.names = NC_UTM$NAME
) %>% nb2lines(coords = NC_centroids$geometry)

k_6 <- knn2nb(
  knn = knearneigh(
    x = NC_centroids,
    k = 6
  ),
  row.names = NC_UTM$NAME
) %>% nb2lines(coords = NC_centroids$geometry)

t1 <- qtm(NC_UTM) + qtm(k_1, title = "k = 1")
t2 <- qtm(NC_UTM) + qtm(k_2, title = "k = 2")
t4 <- qtm(NC_UTM) + qtm(k_4, title = "k = 4")
t6 <- qtm(NC_UTM) + qtm(k_6, title = "k = 6")

tmap_arrange(
  t1, t2,
  t4, t6,
  nrow = 2
)

```



#### Extra credit :)

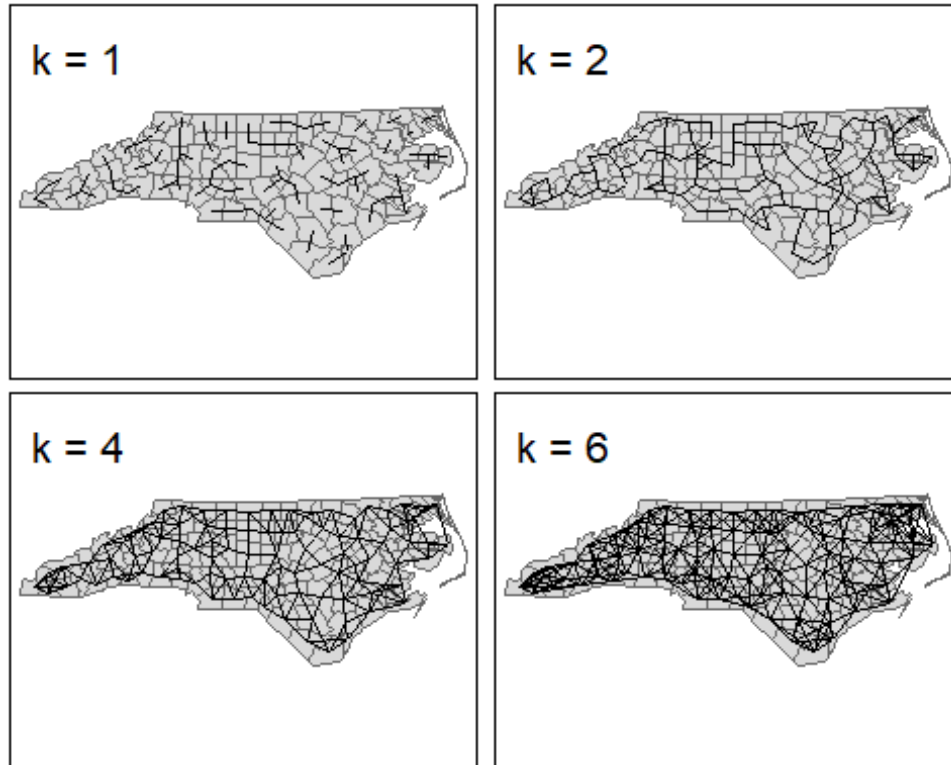
*# This part is super open-ended and meant to develop problem-solving skills with  
# R. I'm a huge fan of purrr, so it's what I used to make these :)*

```
ks <- c(1, 2, 4, 6)

plot_knn <- function(k) {
  knn <- knn2nb(
    knn = knearneigh(
      x = NC_centroids,
      k = k
    ),
    row.names = NC_UTM$NAME
  )
  knn_sf <- knn %>% nb2lines(coords = NC_centroids$geometry)

  qtm(NC_UTM) + qtm(knn_sf, title = paste("k =", k))
}

ks %>%
  purrr::map(plot_knn) %>%
  tmap_arrange()
```



#### Question 4:

- In addition to our “100%” distance threshold (the code above), create plots for:
  - 50% of the maximum distance,
  - 125% of the maximum distance,
  - 150% of the maximum distance.
- Use `tmap` to do this, and arrange all plots in a single image using `tmap_arrange()`. Be sure to label each plot.
- What happened to the number of connections from the 75% to the 125% construct?

```
NC_knn1 <- knn2nb(
  knn = knearneigh(x = NC_centroids,
                   k = 1),
  row.names = NC_UTM$NAME
)

dist <- nbdists(
  nb = NC_knn1,
  coords = NC_centroids$geometry
) %>%
  unlist()

max_dist <- max(dist)

ds <- c(0.75, 1.00, 1.25, 1.50)
```

```

plot_dnn <- function(d) {
  d_threshold <- d * max_dist

  dnn <- dnearneigh(
    x = NC_centroids,
    d1 = 0,
    d2 = d_threshold,
    row.names = NC_UTM$NAME
  )

  dnn_sf <- dnn %>% nb2lines(coords = NC_centroids$geometry)

  qtm(NC_UTM) + qtm(dnn_sf, title = paste0("d = ", as.character(d * 100),
"%"))
}

ds %>%
  purrr::map(plot_dnn) %>%
  tmap_arrange(nrow = 2)

```

**d = 75%**



**d = 100%**



**d = 125%**



**d = 150%**

