library(sf)

## Linking to GEOS 3.9.1, GDAL 3.2.1, PROJ 7.2.1; sf\_use\_s2() is TRUE

library(spdep)

## Loading required package: sp

## Loading required package: spData

## To access larger datasets in this package, install the spDataLarge package  
## with: `install.packages('spDataLarge',  
## repos='https://nowosad.github.io/drat/', type='source')`

library(tmap)  
  
NC <- read\_sf("data/NC\_REGION.shp")  
  
  
NC\_UTM <- st\_transform(NC, crs = st\_crs("EPSG:26917"))

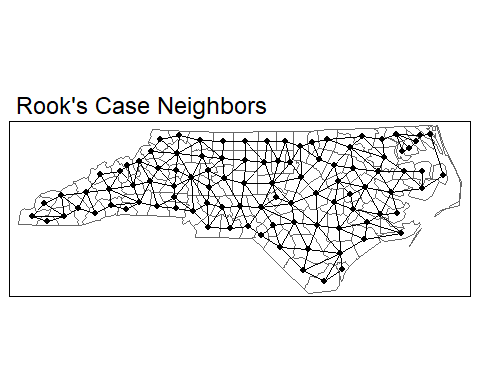
## Question 1:

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"  
 )



## 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

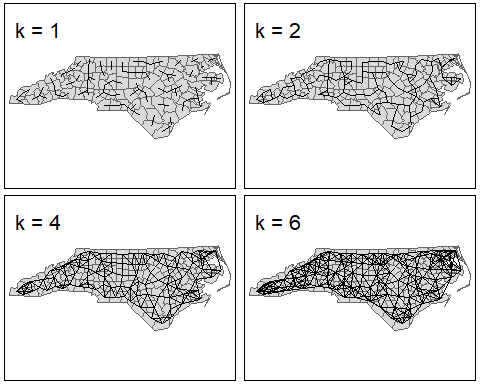
## Question 3:

In addition to , use knn2nb() to create neighbor constructs for , , and . 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”*

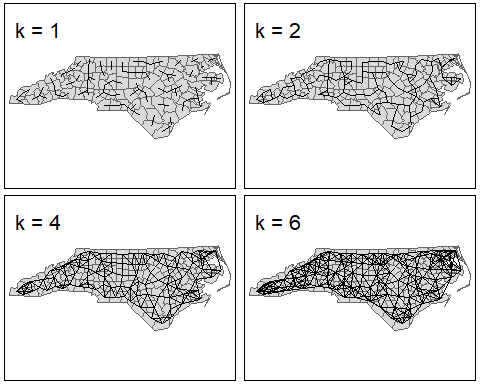
k\_1 <- knn2nb(  
 knn = knearneigh(  
 x = NC\_centroids,  
 k = 1  
 ),  
 row.names = NC\_UTM$NAME  
) %>% nb2lines(coords = NC\_centroids$geometry)  
  
k\_2 <- knn2nb(  
 knn = knearneigh(  
 x = NC\_centroids,  
 k = 2  
 ),  
 row.names = NC\_UTM$NAME  
) %>% nb2lines(coords = NC\_centroids$geometry)  
  
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:

1. 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.

1. Use tmap to do this, and arrange all plots in a single image using tmap\_arrange(). Be sure to label each plot.
2. 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)

