```r
library(sf)
library(spdep)
library(tidyverse)
library(tmap)


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

NC_UTM <- st_transform(
  x = NC,
  crs = st_crs("EPSG:26917")
)
queen_nb <- NC_UTM %>%
  tibble::column_to_rownames("NAME") %>%
  st_as_sf() %>%
  poly2nb(queen = TRUE)


queen_nb_w <- nb2listw(
  neighbours = queen_nb,
  style = "W",
  zero.policy = TRUE
)
```

1. Calculate Moran's I for the following variables in the dataset. List the Moran's I values and p-values for each variable in a single summary table:

- MNEM2000
- MNEM1990
- TOTJOB2000
- TOTJOB1990

Provide your R code for the Moran's I tests (not the results themselves – that's what the table is for).

```r
# It's totally OK if students do this part manually; I just was curious to find
# a way to specify a set of variables to input automatically and then get a
# table as an output.


vars <- c("MNEM2000", "MNEM1990", "TOTJOB2000", "TOTJOB1990", "POP2000", "POP1990")


moran_results <-
  NC_UTM %>%
  st_drop_geometry() %>%
  dplyr::select(any_of(x = vars)) %>%
  map(function(x) {
    moran.test(
      x = x,
      listw = queen_nb_w,
      zero.policy = TRUE,
      alternative = "two.sided"
    )[c("estimate", "p.value")]
  })
```

```
moran_table <- moran_results %>%
  transpose() %>%
  as_tibble() %>%
  unnest_wider(estimate) %>%
  mutate(Variable = vars, .before = `Moran I statistic`)

knitr::kable(moran_table)
```

Variable	Moran I statistic	Expectation	Variance	p.value
MNEM2000	0.3884704	-0.010101	0.0040192	3.238771e-10
MNEM1990	0.4283699	-0.010101	0.0040993	7.468829e-12
TOTJOB2000	0.1631958	-0.010101	0.0035409	0.003587933
TOTJOB1990	0.1642235	-0.010101	0.0037208	0.004264945
POP2000	0.1647821	-0.010101	0.0035306	0.003248218
POP1990	0.1642235	-0.010101	0.0037208	0.004264945

Question 2:

Calculate Moran's I for the `MNEM2000` variable using four different versions of W_{ij}:

- Queen's Case
- Rook's Case
- $k = 4$ nearest neighbors
- Maximum Distance (100% threshold)

Make sure your weights matrices are row standardized. Provide a table that summarized the calculated I, p-values, and average number of connections per county. Discuss any systematic changes you observe in I, p-values, and average links.

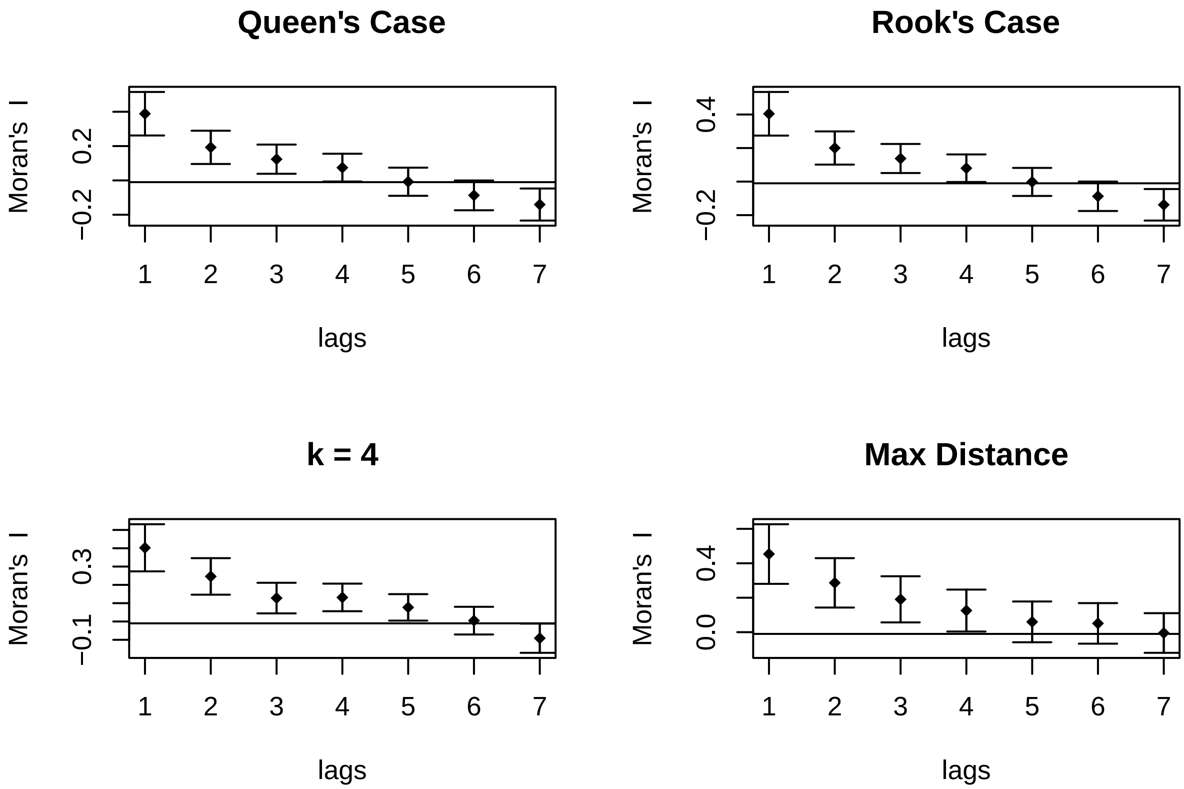
```
NC_centroids <- st_centroid(NC_UTM)

q_w <- NC_UTM %>%
  poly2nb(queen = TRUE)
r_w <- NC_UTM %>%
  poly2nb(queen = FALSE)

k_w <- knn2nb(knearneigh(NC_centroids,
  k = 4
))

max_dist <- knn2nb(
  knearneigh(
    x = NC_centroids,
    k = 1
  )
) %>%
  nbdists(coords = NC_centroids$geometry) %>%
  unlist() %>%
  max()

d_w <- dnearneigh(
  NC_centroids,
  d1 = 0,
```

```r
  d2 = max_dist
)

list_w <- list(
  "Queen's Case" = q_w,
  "Rook's Case" = r_w,
  "k = 4" = k_w,
  "Max Distance" = d_w
)

list_w_weighed <-
  map(list_w, ~ nb2listw(.x, style = "W", zero.policy = TRUE))

moran_w <- list_w_weighed %>%
  map(function(w) {
    moran.test(
      x = NC_UTM$MNEM2000,
      listw = w,
      zero.policy = TRUE,
      alternative = "two.sided"
    )[c("estimate", "p.value")]
  })

neighbors <- list_w_weighed %>%
  map(function(w) {
    w[["neighbours"]] %>%
      unclass() %>%
      map(length) %>%
      unlist() %>%
      mean()
  }) %>%
  enframe() %>%
  rename(Links = value)

moran_table_w <- moran_w %>%
  transpose() %>%
  as_tibble() %>%
  unnest_wider(estimate) %>%
  mutate(Method = names(list_w_weighed), .before = `Moran I statistic`) %>%
  bind_cols(neighbors["Links"])

knitr::kable(moran_table_w)
```

Method	Moran I statistic	Expectation	Variance	p.value	Links
Queen's Case	0.3884704	-0.010101	0.0040192	3.238771e-10	4.9
Rook's Case	0.4044841	-0.010101	0.0042276	1.814863e-10	4.62
k = 4	0.4024844	-0.010101	0.0041371	1.412727e-10	4
Max Distance	0.4536758	-0.010101	0.0074840	8.278861e-08	2.8

Question 3:

```
corr_list <- list_w %>%
  map(~ sp.correlogram(
    neighbours = .x,
    var = NC_UTM$MNEM2000,
    order = 7,
    method = "I",
    zero.policy = TRUE
  ))


par(mfrow = c(2, 2))
corr_plots <- corr_list %>%
  imap(~ {
    plot(.x, main = .y)
  })
```



Queen's Case

Rook's Case

k = 4

Max Distance

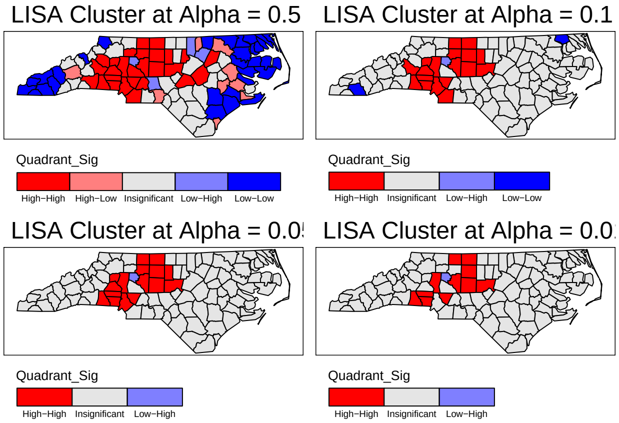
The general relationship is that the I statistic decreases with the number of lags. However, in certain graphs, e.g. $k = 4$, there are small "bumps" where the value increases. This likely indicates that the "centers" of clusters are approximately 3~4 units (counties) away from one another.

Question 4:

4. Adopt the code above to provide LISA maps for the `MNEM1990` variable at an alpha of 0.1, 0.05, and 0.01 using the Queen's case W. Provide a brief explanation of your findings about any local spatial autocorrelation at these more restrictive values of alpha.

The map in this one is slightly different than the example in the "old" R Module 9. I'm using the quadrants generated by the function itself, which is still based on the mean. I'm not quite sure why the results seem to be different.

```r
local_moran_queen <- localmoran(
  x = NC_UTM$MNEM2000,
  listw = queen_nb_w,
  # zero.policy = TRUE,
  alternative = "two.sided"
)


lm_pal <-
  c(
    "High-High" = "red",
    "High-Low" = "#FF000080",
    "Insignificant" = "gray90",
    "Low-High" = "#0000FF80",
    "Low-Low" = "blue"
  )



quadr <- attr(local_moran_queen, "quadr")[, 2]

NC_localmoran <- NC_UTM

NC_localmoran$Quadrant <- quadr

NC_localmoran$p.value <- local_moran_queen[, 5]

sig_level <- 0.5

plot_LISA_sig <- function(sig) {
  NC_localmoran <- NC_localmoran %>%
    mutate(
      Quadrant_Sig = if_else(p.value <= sig,
        true = as.character(Quadrant),
        false = "Insignificant"
      ) %>%
        as.factor()
    )
  t <- tm_shape(NC_localmoran) +
    tm_polygons(
      col = "Quadrant_Sig",
      palette = lm_pal,
      legend.is.portrait = F,
      border.col = "black"
    ) +
    tm_layout(
      main.title = paste0("LISA Cluster at Alpha = ", sig),
      legend.outside = T,
      legend.outside.position = "bottom"
    )
  return(t)
}

LISA_list <- map(c(0.5, 0.1, 0.05, 0.01), plot_LISA_sig)

tmap_arrange(LISA_list)
```


LISA Cluster at Alpha = 0.5
Quadrant_Sig
High−High
High−Low
Insignificant
Low−High
Low−Low
LISA Cluster at Alpha = 0.1
Quadrant_Sig
High−High
Insignificant
Low−High
Low−Low
LISA Cluster at Alpha = 0.0
Quadrant_Sig
High−High
Insignificant
Low−High
LISA Cluster at Alpha = 0.0
Quadrant_Sig
High−High
Insignificant
Low−High