Homework 3

Peter Tran

```
library(ape)
library(sf)
library(spdep)
library(tidyverse)
library(tigris)
library(tmap)
library(GISTools)
```

First we'll pull in both the COVID deaths and the population counts for each county. We only care about Missouri, so we'll filter to only Missouri and we'll also remove the statewide unallocated row in each dataset.

```
deaths <- read_csv("../../data/raw/hw/covid_deaths_usafacts.csv") %>%
  filter(StateFIPS == 29 & countyFIPS != 0)
```

```
## Rows: 3193 Columns: 772
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (3): County Name, State, StateFIPS
## dbl (769): countyFIPS, 2020-01-22, 2020-01-23, 2020-01-24, 2020-01-25, 2020-...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
population <- read_csv("../../data/raw/hw/covid_county_population_usafacts.csv") %>%
  filter(State == "MO" & countyFIPS != 0)
```

```
## Rows: 3195 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): County Name, State
## dbl (2): countyFIPS, population
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Now we'll read in the shapefiles for Missouri through tigris.

```
mo_shapefiles <- counties(29)
```

```
##   |                                                                              |
```

Let's join all those together now.

```
mo <- mo_shapefiles %>% merge(population, by.x = "GEOID", by.y = "countyFIPS") %>%
  merge(deaths, by.x = "GEOID", by.y = "countyFIPS")
```

Ok, these deaths are cumulative, so we'll need to subtract the first day from the last day for each year of interest.

```
mo$`2020_deaths` <- mo$`2020-12-31`
mo$`2021_deaths` <- mo$`2021-12-31` - mo$`2020-12-31`

mo <- mo %>%
  dplyr::select(colnames(.) %>% discard(function(x) grepl("2020-", x, fixed = T))) %>%
  dplyr::select(colnames(.) %>% discard(function(x) grepl("2021-", x, fixed = T))) %>%
  dplyr::select(colnames(.) %>% discard(function(x) grepl("2022-", x, fixed = T)))
```

Now we can calculate the rates per 100k.

```
mo$`2020_rates` <- mo$`2020_deaths` / mo$population * 100000
mo$`2021_rates` <- mo$`2021_deaths` / mo$population * 100000
```

Our data is all set up, now we can calculate Moran's I. Let's start by using the spdep package to calculate Moran's I for both 2020 as well as 2021 rates, using the default weight matrix (derived from the queen adjacency matrix). We'll use the simulation approach.

```
mo_nb <- poly2nb(mo)
mo_default_lw <- nb2listw(mo_nb)

moran.mc(mo$`2020_rates`, mo_default_lw, 100000)
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  mo$`2020_rates`
## weights: mo_default_lw
## number of simulations + 1: 100001
##
## statistic = 0.078724, observed rank = 92612, p-value = 0.07389
## alternative hypothesis: greater
```

```
moran.mc(mo$`2021_rates`, mo_default_lw, 100000)
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  mo$`2021_rates`
## weights: mo_default_lw
## number of simulations + 1: 100001
##
## statistic = 0.23527, observed rank = 99988, p-value = 0.00013
## alternative hypothesis: greater
```

It looks like we don't have strong evidence of spatial autocorrelation for the 2020 rates, but we do for the 2021 rates!

Now let's do the same using the inverse distance matrix with the ape package.

```
mo_distance_matrix = as.matrix(dist(coordinates(as_Spatial(mo))))
mo_distance_matrix_inverted = 1 / mo_distance_matrix
diag(mo_distance_matrix_inverted) = 0
mo_distance_matrix_inverted =
  mo_distance_matrix_inverted / apply(mo_distance_matrix_inverted, 1, sum)

Moran.I(mo$`2020_rates`, mo_distance_matrix_inverted)
```

```
## $observed
## [1] 0.01109355
##
## $expected
## [1] -0.00877193
##
## $sd
## [1] 0.01011958
##
## $p.value
## [1] 0.04963756
```

```
Moran.I(mo$`2021_rates`, mo_distance_matrix_inverted)
```

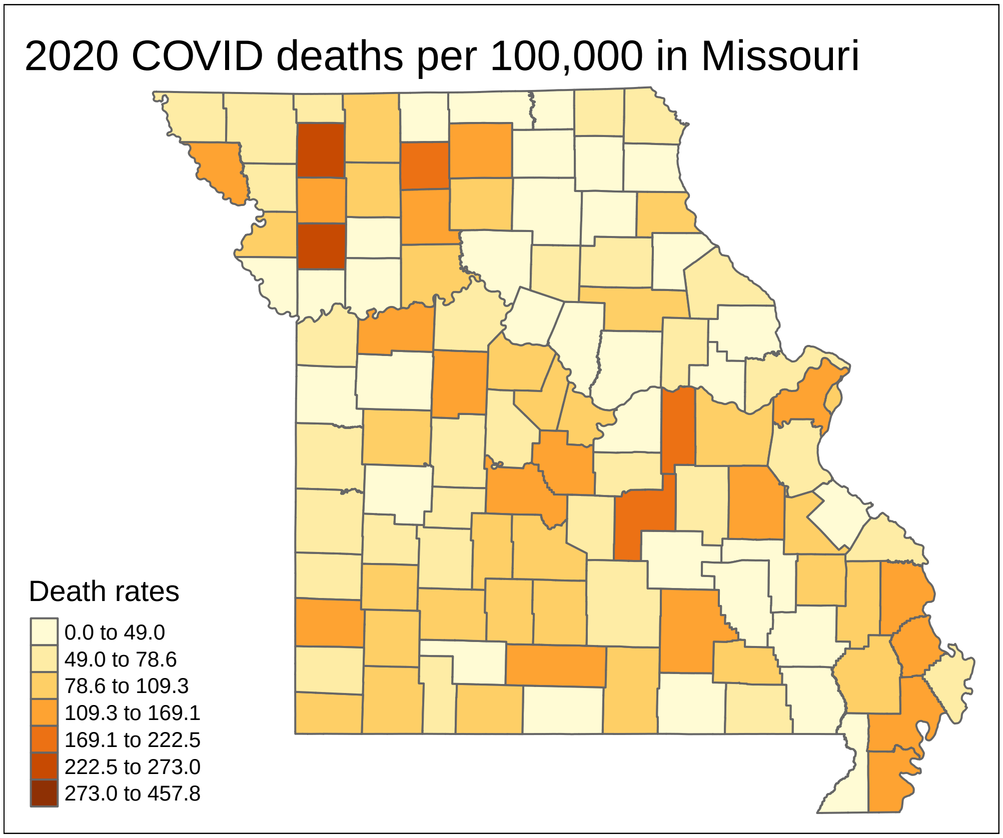
```
## $observed
## [1] 0.02865232
##
## $expected
## [1] -0.00877193
##
## $sd
## [1] 0.01019059
##
## $p.value
## [1] 0.0002402538
```

We see strong evidence in both rates, though it is much stronger in the 2021 rates. Interestingly the strength of the spatial autocorrelation in the 2021 rates is much lower in the inverse distance matrix method compared to the distance matrix derived from the queen adjacency matrix.

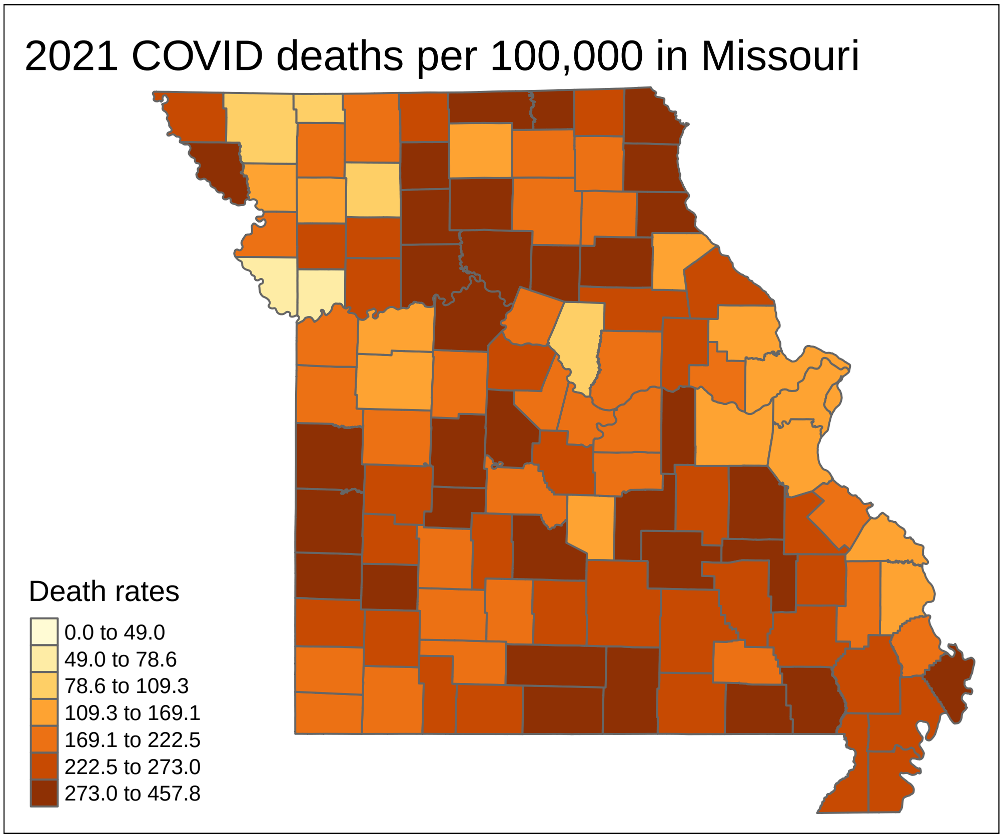
Let's plot choropleth maps for both years of rates to put a picture to these Moran's Is.

```
rate_breaks <- c(0, round(quantileCuts(c(mo$`2020_rates`, mo$`2021_rates`), 7), 1),
                 max(c(mo$`2020_rates`, mo$`2021_rates`)))

tm_shape(mo) +
  tm_polygons("2020_rates", title = "Death rates", breaks = rate_breaks) +
  tm_layout(title = "2020 COVID deaths per 100,000 in Missouri",
            legend.position = c("left", "bottom"),
            inner.margins = c(0.025, 0.15, 0.1, 0.025))
```



```
tm_shape(mo) +
  tm_polygons("2021_rates", title = "Death rates", breaks = rate_breaks) +
  tm_layout(title = "2021 COVID deaths per 100,000 in Missouri",
            legend.position = c("left", "bottom"),
            inner.margins = c(0.025, 0.15, 0.1, 0.025))
```



It's really hard to judge it visually, but it does feel like there is more "clumping" in the 2021 rates compared to the 2020 rates. This may be more true when trying to look at it from an adjacency perspective rather than a true distance perspective.