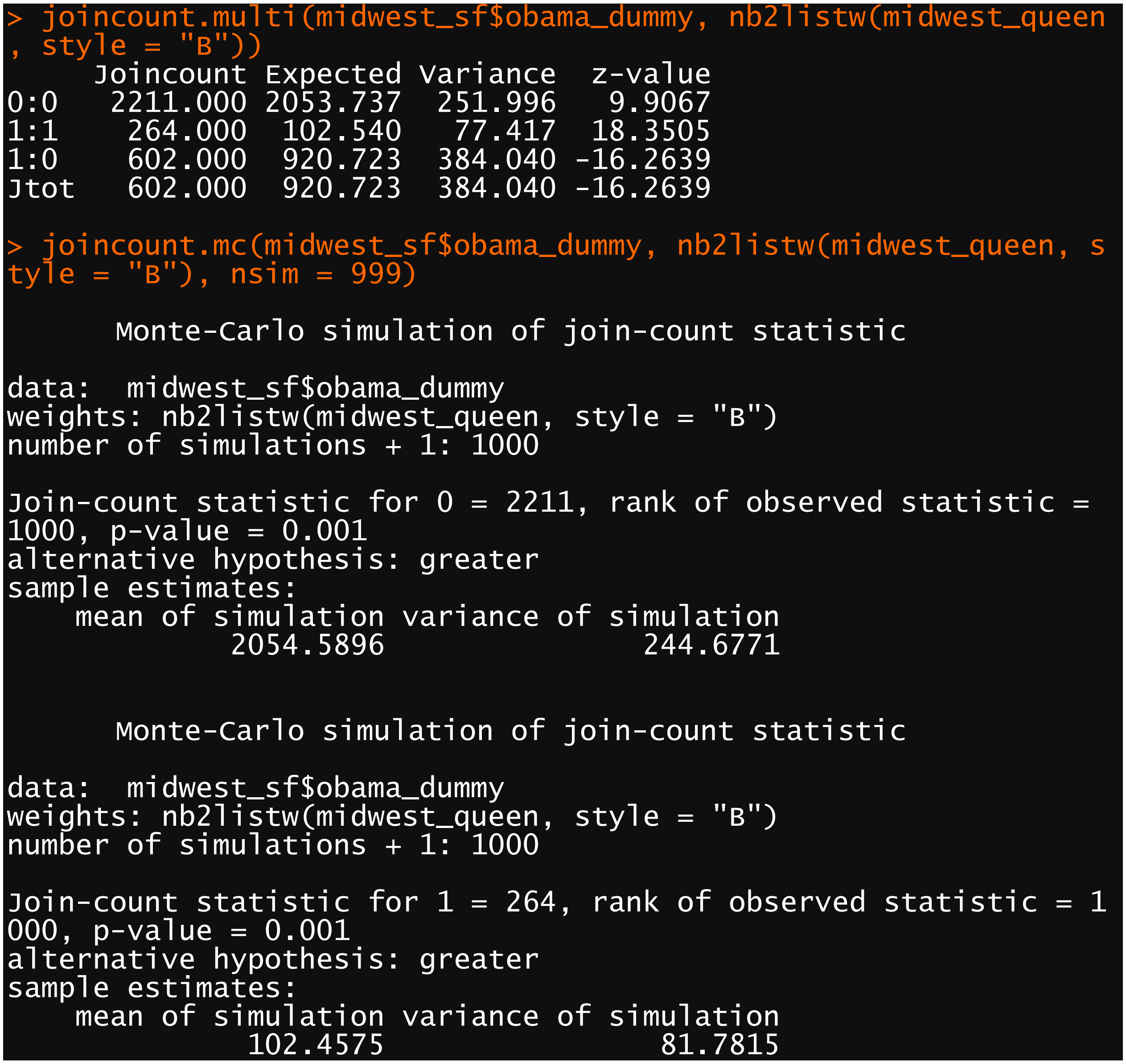
Sean Connelly
UPP 465
Professor Tilahun
3/30/20

Spatial Statistics – Global Measures of Spatial Autocorrelation

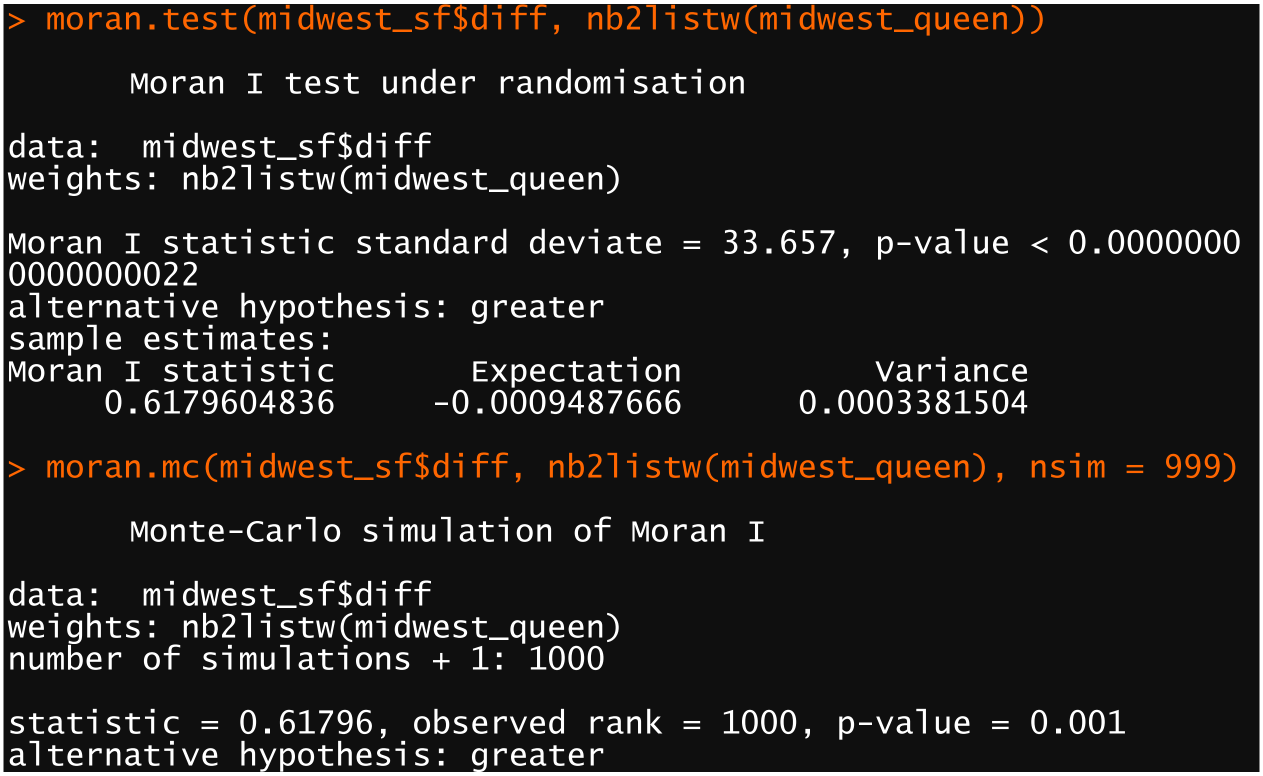
The global autocorrelation measures calculated below use 2012 presidential election vote share data gathered from 1,055 counties in the Midwest.

1. Join counts



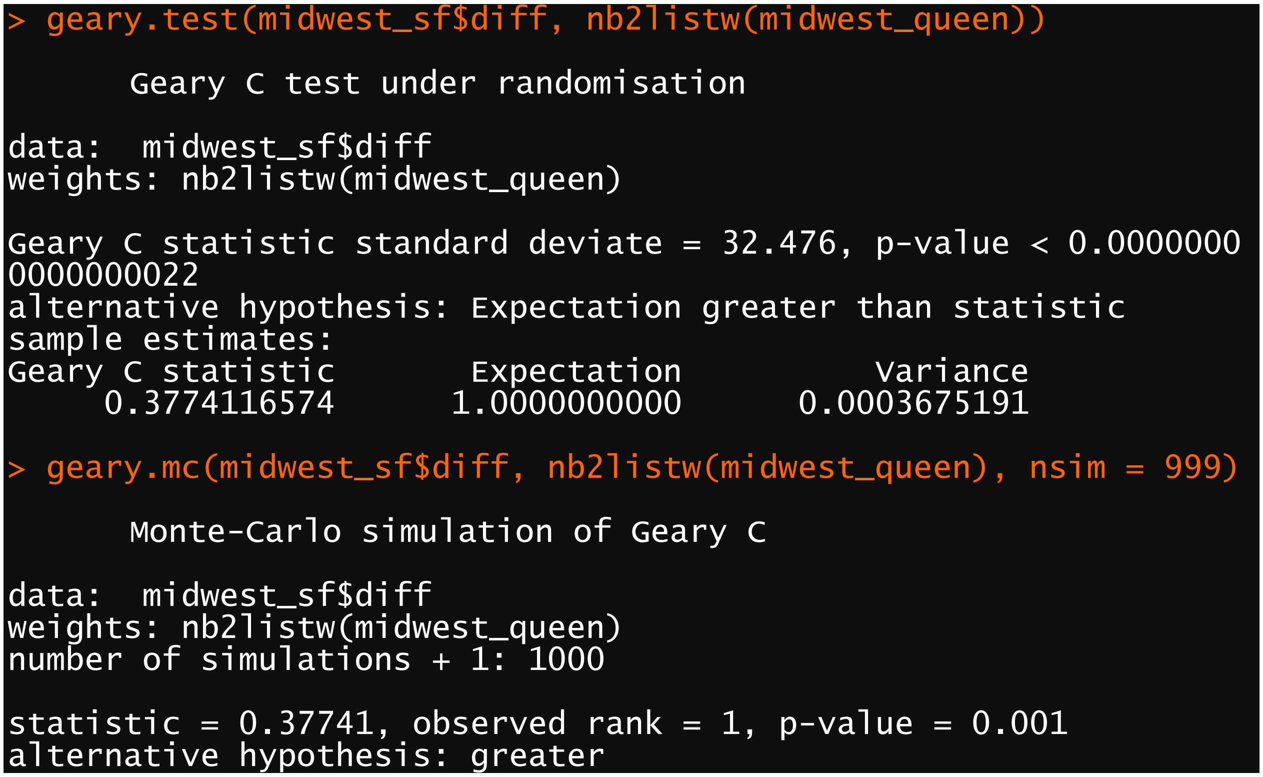
```
> joincount.multi(midwest_sf$obama_dummy, nb2listw(midwest_queen
, style = "B"))
     Joincount Expected Variance  z-value
0:0   2211.000 2053.737  251.996   9.9067
1:1    264.000  102.540   77.417  18.3505
1:0    602.000  920.723  384.040 -16.2639
Jtot   602.000  920.723  384.040 -16.2639

> joincount.mc(midwest_sf$obama_dummy, nb2listw(midwest_queen, s
tyle = "B"), nsim = 999)

	Monte-Carlo simulation of join-count statistic

data:  midwest_sf$obama_dummy 
weights: nb2listw(midwest_queen, style = "B") 
number of simulations + 1: 1000 

Join-count statistic for 0 = 2211, rank of observed statistic =
1000, p-value = 0.001
alternative hypothesis: greater
sample estimates:
    mean of simulation variance of simulation 
             2054.5896                244.6771 


	Monte-Carlo simulation of join-count statistic

data:  midwest_sf$obama_dummy 
weights: nb2listw(midwest_queen, style = "B") 
number of simulations + 1: 1000 

Join-count statistic for 1 = 264, rank of observed statistic = 1
000, p-value = 0.001
alternative hypothesis: greater
sample estimates:
    mean of simulation variance of simulation 
              102.4575                81.7815 
```

2. Moran’s I



```
> moran.test(midwest_sf$diff, nb2listw(midwest_queen))

	Moran I test under randomisation

data:  midwest_sf$diff  
weights: nb2listw(midwest_queen)    

Moran I statistic standard deviate = 33.657, p-value < 0.0000000
0000000022
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance 
     0.6179604836     -0.0009487666      0.0003381504 

> moran.mc(midwest_sf$diff, nb2listw(midwest_queen), nsim = 999)

	Monte-Carlo simulation of Moran I

data:  midwest_sf$diff 
weights: nb2listw(midwest_queen)  
number of simulations + 1: 1000 

statistic = 0.61796, observed rank = 1000, p-value = 0.001
alternative hypothesis: greater
```

3. Geary’s C



```
> geary.test(midwest_sf$diff, nb2listw(midwest_queen))

	Geary C test under randomisation

data:  midwest_sf$diff 
weights: nb2listw(midwest_queen) 

Geary C statistic standard deviate = 32.476, p-value < 0.0000000
0000000022
alternative hypothesis: Expectation greater than statistic
sample estimates:
Geary C statistic       Expectation          Variance 
     0.3774116574      1.0000000000      0.0003675191 

> geary.mc(midwest_sf$diff, nb2listw(midwest_queen), nsim = 999)

	Monte-Carlo simulation of Geary C

data:  midwest_sf$diff 
weights: nb2listw(midwest_queen)  
number of simulations + 1: 1000 

statistic = 0.37741, observed rank = 1, p-value = 0.001
alternative hypothesis: greater
```

4. The join-counts, Moran’s I, and Geary’s C test results all indicate that there is evidence of positive autocorrelation in the vote data. The Monte-Carlo simulations of the joint-count statistic suggest that there are far more “1:1” (Obama-Obama) and “0:0” (Romney-Romney) counts than one would expect to see under a pattern of spatial randomness. Moran’s I is positive and significant, providing further evidence of clustering. Finally, Geary’s C is less than one and significant, which is also a sign that similar vote shares are spatially close together.

```
# Load libraries ---------------------------------------------------------
#make sure the below libraries are installed on your machine. Use
install.packages("LIBRARYNAME")
rm(list = ls()) #clear objects in memory
# Set working directory
setwd(paste0(here::here(), "/08_Global_Measures"))
# Load additional libraries
pacman::p_load(tidyverse, # for basic data manipulation, visualization
scales, # for formatting number output
patchwork, # for arranging ggplots in grids
sf, # simple features for spatial
nngeo, # nearest neighbors
summarytools, # for checking data frame characteristics
here, # for relative file paths
knitr, # for tables
kableExtra, # table styling
janitor) # for cleaning and tabulations
#load your spatial libraries
library(sp)
library(raster)
library(spdep)
options(stringsAsFactors = FALSE, scipen = 999)
# ***************************************************************************
# START OF ASSIGNMENT
# ***************************************************************************
# Assignment (Due April 6, 2020) ----------------------------------------
# Read in RDS
midwest <- readRDS("Raw_Data/Midwest_2012_election.RDS")
# Convert to simple features
midwest_sf <- st_as_sf(midwest, crs = 4269)
# Define neighbors
# Queen
midwest_queen <- poly2nb(midwest_sf)
# # Rook
# midwest_rook <- poly2nb(midwest_sf, queen = FALSE)
#
# # Identify the difference
# midwest_diff <- diffnb(midwest_queen, midwest_rook)
#
# # Get the centroids of the counties
# midwest_cent <- coordinates(midwest)
#
# #plot the three neighbor definitions
# par(mfrow = c(1, 3))
# plot(midwest, border = 8)
# plot(midwest_queen, midwest_cent, add = TRUE)
# title(main = paste("Midwest Counties Queen"))
#
# plot(midwest, border = 8)
# plot(midwest_rook, midwest_cent, add = TRUE)
# title(main = paste("Midwest Counties Rook"))
#
# plot(midwest, border = 8)
# plot(midwest_diff, midwest_cent, add = TRUE, col = 2)
# title(main = paste("Midwest Counties Difference"))
# Create dummy variable
```

```
midwest_sf <- midwest_sf %>%
mutate(obama_dummy = ifelse(WINNER == "Obama", 1, 0) %>%
as_factor())
# Compute global autocorrelation measures using the Midwest data
#1. Use join count based on a variable that takes 1 when Obamawon and 0
otherwise
joincount.multi(midwest_sf$obama_dummy, nb2listw(midwest_queen, style = "B"))
joincount.mc(midwest_sf$obama_dummy, nb2listw(midwest_queen, style = "B"),
nsim = 999)
#2. Use Moran's I
moran.test(midwest_sf$diff, nb2listw(midwest_queen))
moran.mc(midwest_sf$diff, nb2listw(midwest_queen), nsim = 999)
#3. Use Geary's C
geary.test(midwest_sf$diff, nb2listw(midwest_queen))
geary.mc(midwest_sf$diff, nb2listw(midwest_queen), nsim = 999)
#4. Interpret your results
```