Part 1 3 Variables

This is just for the sake of completeness.

```
library(sp)
library(spmodel)
```

```
## Warning: package 'spmodel' was built under R version 4.3.3
```

```
library(sf)
```

```
## Warning: package 'sf' was built under R version 4.3.3
```

```
## Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE
```

```
library(spdep)
```

```
## Warning: package 'spdep' was built under R version 4.3.3
```

```
## Loading required package: spData
```

```
## Warning: package 'spData' was built under R version 4.3.3
```

```
## 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)
```

```
## Warning: package 'tmap' was built under R version 4.3.3
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.3.3
```

```
## corrplot 0.95 loaded
```

```
library(readr)
library(tidyverse)
```

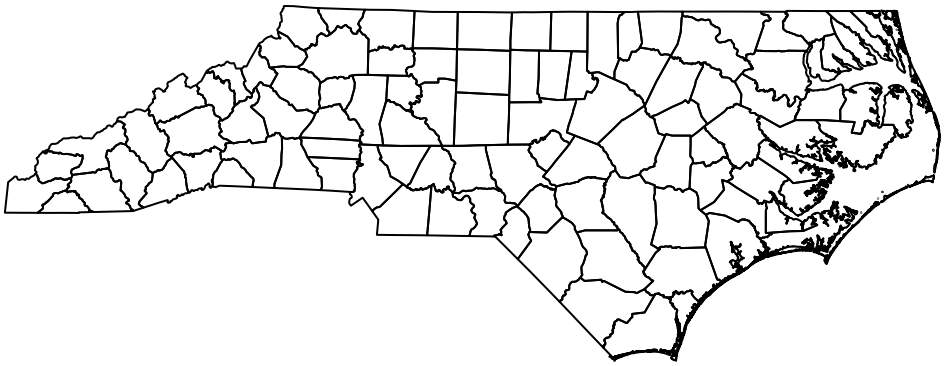
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v purrr     1.0.2
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(tinytex)
library(ggcorrplot)

## Load in the demographic data and county data for North Carolina
ncdata=read.csv("NCcounty2024data.csv",header=T)
load("nc_counties.Rdata")

## Plot the counties
plot(nc_counties)
```



```
## Names of the counties
nc_counties[[7]]
```

```
##   [1] "Alamance"     "Alexander"    "Alleghany"    "Anson"        "Ashe"
##   [6] "Avery"        "Beaufort"     "Bertie"       "Bladen"       "Brunswick"
##  [11] "Buncombe"     "Burke"        "Cabarrus"     "Caldwell"     "Camden"
##  [16] "Carteret"     "Caswell"      "Catawba"      "Chatham"      "Cherokee"
##  [21] "Chowan"       "Clay"         "Cleveland"    "Columbus"     "Craven"
##  [26] "Cumberland"   "Currituck"    "Dare"         "Davidson"     "Davie"
##  [31] "Duplin"       "Durham"       "Edgecombe"    "Forsyth"      "Franklin"
##  [36] "Gaston"       "Gates"        "Graham"       "Granville"    "Greene"
##  [41] "Guilford"     "Halifax"      "Harnett"      "Haywood"      "Henderson"
##  [46] "Hertford"     "Hoke"         "Hyde"         "Iredell"      "Jackson"
##  [51] "Johnston"     "Jones"        "Lee"          "Lenoir"       "Lincoln"
##  [56] "Macon"        "Madison"      "Martin"       "McDowell"     "Mecklenburg"
##  [61] "Mitchell"     "Montgomery"   "Moore"        "Nash"         "New Hanover"
##  [66] "Northampton"  "Onslow"       "Orange"       "Pamlico"      "Pasquotank"
##  [71] "Pender"       "Perquimans"   "Person"       "Pitt"         "Polk"
##  [76] "Randolph"     "Richmond"     "Robeson"      "Rockingham"   "Rowan"
```

```
##  [81] "Rutherford"   "Sampson"      "Scotland"     "Stanly"       "Stokes"
##  [86] "Surry"        "Swain"        "Transylvania" "Tyrrell"      "Union"
##  [91] "Vance"        "Wake"         "Warren"       "Washington"   "Watauga"
##  [96] "Wayne"        "Wilkes"       "Wilson"       "Yadkin"       "Yancey"
```

```
## Obtain neighborhoods given the polygons
nc_neigh=poly2nb(nc_counties)
```

```
# Convert to sf object before merge
nc_counties_sf <- st_as_sf(nc_counties)

# Make sure the name matches
nc_counties_sf$County <- nc_counties_sf$NAME_2

# Now merge
nc_full <- left_join(nc_counties_sf, ncdata, by = "County")

#Neighborhoods for our analyses

# Define spatial neighbors using Queen's Contiguity
nc_nb <- poly2nb(nc_counties_sf, queen = TRUE)


# Convert to a spatial weights list object
nc_listw <- nb2listw(nc_nb, style = "W")
```

For Food Insecurity:

```
# Moran's I Test (Global Spatial Autocorrelation)
moran_test_Food <- moran.test(nc_full$Food.Insecurity.2022, nc_listw)

# Geary's C Test (Local Spatial Autocorrelation)
geary_test_Food <- geary.test(nc_full$Food.Insecurity.2022, nc_listw)

# Print results
print(moran_test_Food)
```

```
##
##  Moran I test under randomisation
##
## data:  nc_full$Food.Insecurity.2022
## weights: nc_listw
##
## Moran I statistic standard deviate = 5.9675, p-value = 1.205e-09
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.378017774      -0.010101010       0.004230048
```

```
print(geary_test_Food)
```

```
##
```

```
##  Geary C test under randomisation
##
## data:  nc_full$Food.Insecurity.2022
## weights: nc_listw
##
## Geary C statistic standard deviate = 5.0315, p-value = 2.433e-07
## alternative hypothesis: Expectation greater than statistic
## sample estimates:
## Geary C statistic       Expectation          Variance
##       0.650030681       1.000000000       0.004837982
```

```
# Moran's I Test (Global Spatial Autocorrelation)
moran_test_Overdose <- moran.test(nc_full$Overdose.Deaths.Per.100.000, nc_listw)

# Geary's C Test (Local Spatial Autocorrelation)
geary_test_Overdose <- geary.test(nc_full$Overdose.Deaths.Per.100.000, nc_listw)

# Print results
print(moran_test_Overdose)
```

```
##
##  Moran I test under randomisation
##
## data:  nc_full$Overdose.Deaths.Per.100.000
## weights: nc_listw
##
## Moran I statistic standard deviate = 3.2762, p-value = 0.000526
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.202493543      -0.010101010       0.004210749
```

```
print(geary_test_Overdose)
```

```
##
##  Geary C test under randomisation
##
## data:  nc_full$Overdose.Deaths.Per.100.000
## weights: nc_listw
##
## Geary C statistic standard deviate = 3.4488, p-value = 0.0002816
## alternative hypothesis: Expectation greater than statistic
## sample estimates:
## Geary C statistic       Expectation          Variance
##       0.757639361       1.000000000       0.004938503
```

```
# Moran's I Test (Global Spatial Autocorrelation)
moran_test_Poverty <- moran.test(nc_full$Percent.Children.in.Poverty, nc_listw)

# Geary's C Test (Local Spatial Autocorrelation)
geary_test_Poverty <- geary.test(nc_full$Percent.Children.in.Poverty, nc_listw)
```

```r
# Print results
print(moran_test_Poverty)
```

```
##
##  Moran I test under randomisation
##
## data:  nc_full$Percent.Children.in.Poverty
## weights: nc_listw
##
## Moran I statistic standard deviate = 6.5275, p-value = 3.343e-11
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.417023892      -0.010101010       0.004281657
```

```r
print(geary_test_Poverty)
```

```
##
##  Geary C test under randomisation
##
## data:  nc_full$Percent.Children.in.Poverty
## weights: nc_listw
##
## Geary C statistic standard deviate = 6.2864, p-value = 1.625e-10
## alternative hypothesis: Expectation greater than statistic
## sample estimates:
## Geary C statistic       Expectation          Variance
##       0.575068377       1.000000000       0.004569172
```

```r
moran_test_Uninsured <- moran.test(nc_full$Uninsured.Residents.2020, nc_listw)

# Geary's C Test (Local Spatial Autocorrelation)
geary_test_Uninsured <- geary.test(nc_full$Uninsured.Residents.2020, nc_listw)

# Print results
print(moran_test_Uninsured)
```

```
##
##  Moran I test under randomisation
##
## data:  nc_full$Uninsured.Residents.2020
## weights: nc_listw
##
## Moran I statistic standard deviate = 4.2552, p-value = 1.044e-05
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.263314499      -0.010101010       0.004128678
```

```r
print(geary_test_Uninsured)
```

```
##
##  Geary C test under randomisation
##
## data:  nc_full$Uninsured.Residents.2020
## weights: nc_listw
##
## Geary C statistic standard deviate = 3.9943, p-value = 3.244e-05
## alternative hypothesis: Expectation greater than statistic
## sample estimates:
## Geary C statistic       Expectation          Variance
##        0.70740365        1.00000000       0.00536597
```