---

title: “GES LAB 7” author: “Justin Johnson” date: “04/2020” output: html_document —

This is an R Markdown (http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

Part 1: Initial Setup

```
#setup
knitr::opts_knit$set(root.dir = "C:/Users/justi/Downloads/School/GES 687/LAB 7/Lab 7 R Code")
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.4
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.5
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## Warning: package 'dplyr' was built under R version 4.0.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidycensus)
```

```
## Warning: package 'tidycensus' was built under R version 4.0.4
```

```
library(ggplot2)
#theme_set(theme_bw()) uncomment to use the bw theme in all ggplot maps
library(sf)
```

```
## Warning: package 'sf' was built under R version 4.0.4
```

```
## Linking to GEOS 3.8.0, GDAL 3.0.4, PROJ 6.3.1
```

```
library(sp)
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(janitor)
```

```
## Warning: package 'janitor' was built under R version 4.0.4
```

```
##
## Attaching package: 'janitor'
```

```
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(readr)
```

Bringing Full Census Data List Into R

```
acs_variable_list = load_variables(2019,"acs5", cache= TRUE)

write.csv(acs_variable_list,'acs_variable_list_2018.csv', row.names = FALSE)
```

Downloading State and County Variables

```
Maryland5year = get_acs(geography = "tract", state = "MD", year=2019, survey="acs5",
                 variables = c("Total Pop" = "B01003_001",
"Black Pop" = "B01001B_001",
                               "medHouseprice" = "B25077_001"),
                 geometry = TRUE,
                 output = "wide") %>% clean_names()
```

```
## Getting data from the 2015-2019 5-year ACS
```

```
## Downloading feature geometry from the Census website.  To cache shapefiles for use in future
sessions, set `options(tigris_use_cache = TRUE)`.
```

```
##
  |
  |                                                                      |   0%
  |
  |=                                                                     |   2%
  |
  |===                                                                   |   4%
  |
  |=====                                                                 |   6%
  |
  |======                                                                |   8%
  |
  |=======                                                               |  10%
  |
  |=========                                                             |  12%
  |
  |=========                                                             |  13%
  |
  |===========                                                           |  16%
  |
  |=============                                                         |  18%
  |
  |==============                                                        |  20%
  |
  |================                                                      |  21%
  |
  |==================                                                    |  25%
  |
  |====================                                                  |  27%
  |
  |=====================                                                 |  29%
  |
  |======================                                                |  30%
  |
  |========================                                              |  33%
  |
  |=========================                                             |  35%
  |
  |===========================                                           |  37%
  |
  |============================                                          |  38%
  |
  |==============================                                        |  41%
  |
  |================================                                      |  43%
  |
  |==================================                                    |  46%
  |
  |===================================                                   |  49%
  |
  |=====================================                                 |  51%
  |
  |======================================                                |  55%
```

```
  |
  |=======================================                               |  57%
  |
  |========================================                              |  58%
  |
  |=========================================                             |  60%
  |
  |==========================================                            |  63%
  |
  |===========================================                           |  66%
  |
  |============================================                          |  68%
  |
  |=============================================                         |  70%
  |
  |==============================================                        |  71%
  |
  |================================================                      |  74%
  |
  |=================================================                     |  78%
  |
  |==================================================                    |  79%
  |
  |====================================================                  |  82%
  |
  |=====================================================                 |  85%
  |
  |======================================================                |  88%
  |
  |=======================================================               |  90%
  |
  |========================================================              |  91%
  |
  |==========================================================            |  95%
  |
  |===========================================================           |  96%
  |
  |============================================================          |  98%
  |
  |============================================================          |  99%
  |
  |============================================================| 100%
```

```
Arundeltractdata2019 = get_acs(geography = "tract", year=2019, state = "MD", survey="acs5", coun
ty = "Anne Arundel",
                 variables = c("Total Pop" = "B01003_001",
"Black Pop" = "B01001B_001",
                               "medHouseprice" = "B25077_001"),
                 geometry = TRUE,
                 output = "wide") %>% clean_names()
```

```
## Getting data from the 2015-2019 5-year ACS
## Downloading feature geometry from the Census website.  To cache shapefiles for use in future sessions, set `options(tigris_use_cache = TRUE)`.
```

```
class(Arundeltractdata2019)
```

```
## [1] "sf"         "data.frame"
```

Part 2: Questions 1-10

1. What is the class of the object? (2 points)

```
class(Maryland5year)
```

```
## [1] "sf"         "data.frame"
```

```
# the class of the object is specified as a shapefile data frame.
# The geometry type of the object is classified as a multi-polygon
```

2. Is the data set you downloaded in question 1 projected or unprojected? Explain in words how you know this. (2 points)

```
#The data that was downloaded belongs to a projected dataset. This is known because when the ful
l, unaltered data is plotted, the polygons are mapped to coordinates, indicating that a CRS is i
ncluded with the dataset, which could only be possible if the dataset was projected.
```

3. Which census tract has the highest total population estimate?

```
max(Maryland5year$total_pop_e)
```

```
## [1] 14369
```

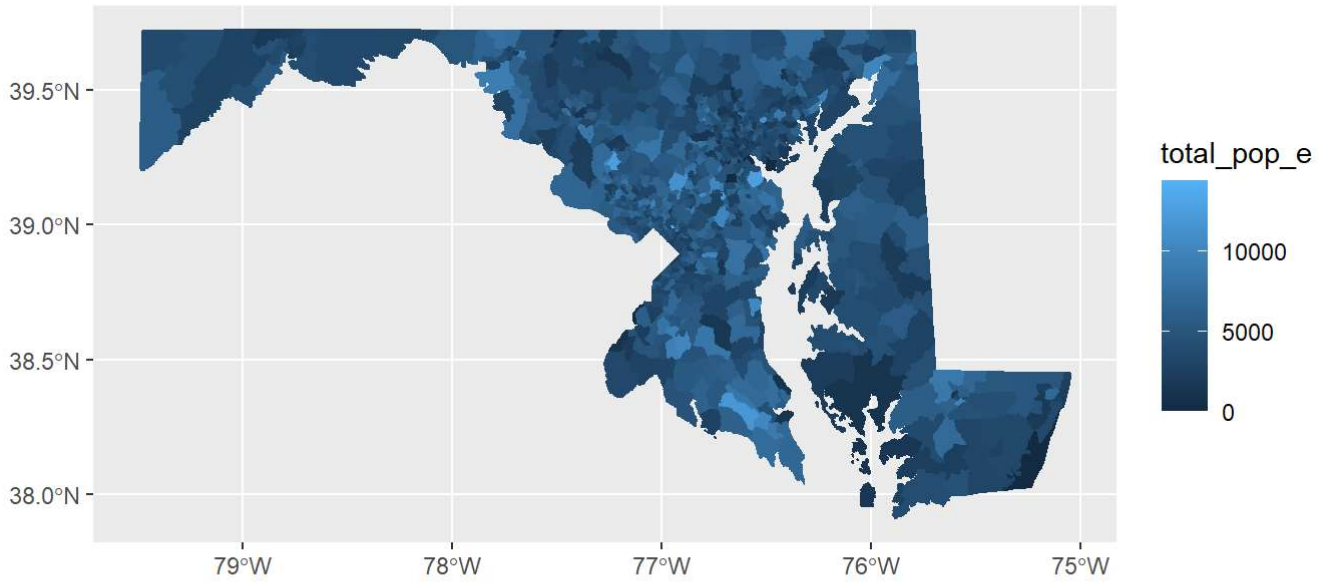
```
#The data shows that the Maryland census tract with the highest population is in the Census trac
t 8072 which is in Prince Georges County
```

**4. How many census tracts have a current total population estimate above 10,000? Plot only these census tracts. (2 points)*

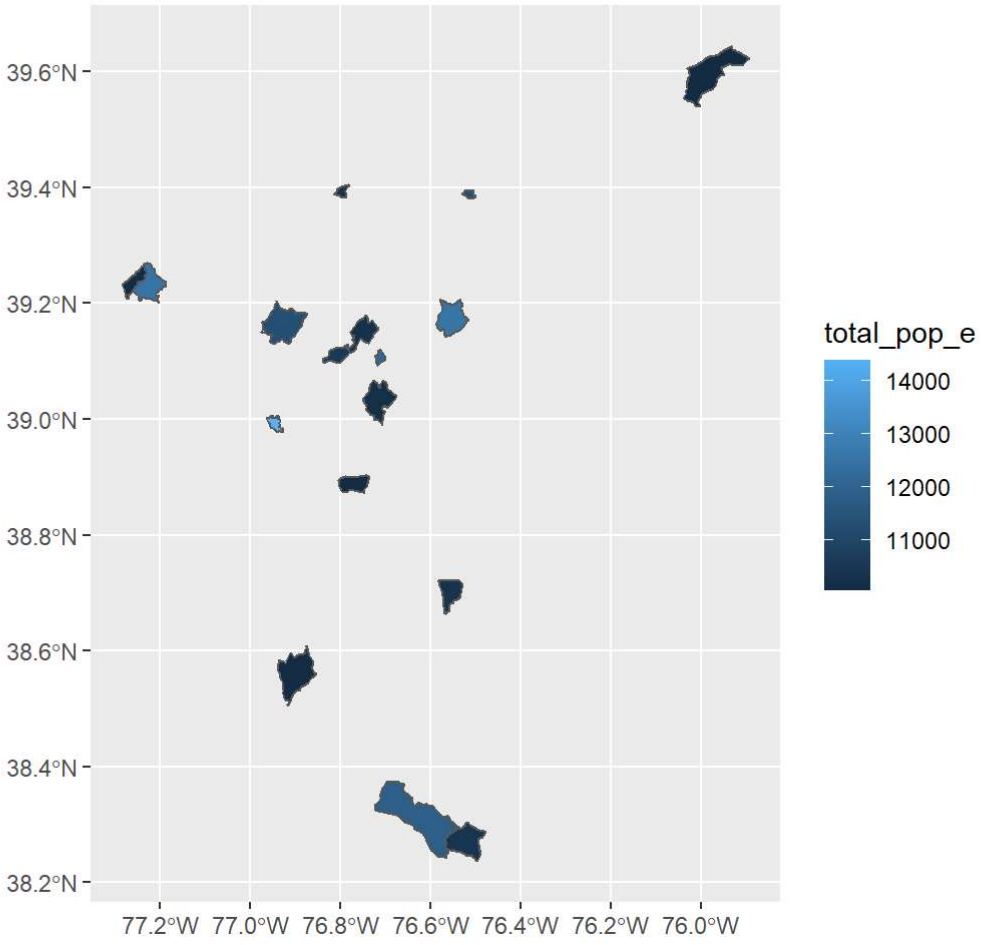
```
sum(Maryland5year$total_pop_e > 10000)
```

```
## [1] 17
```

```
#There are 17 Census tracts in the state of Maryland with more than 10,000 people



#Full Map

ggplot(Maryland5year) + geom_sf(aes(fill = total_pop_e, color = total_pop_e))
```

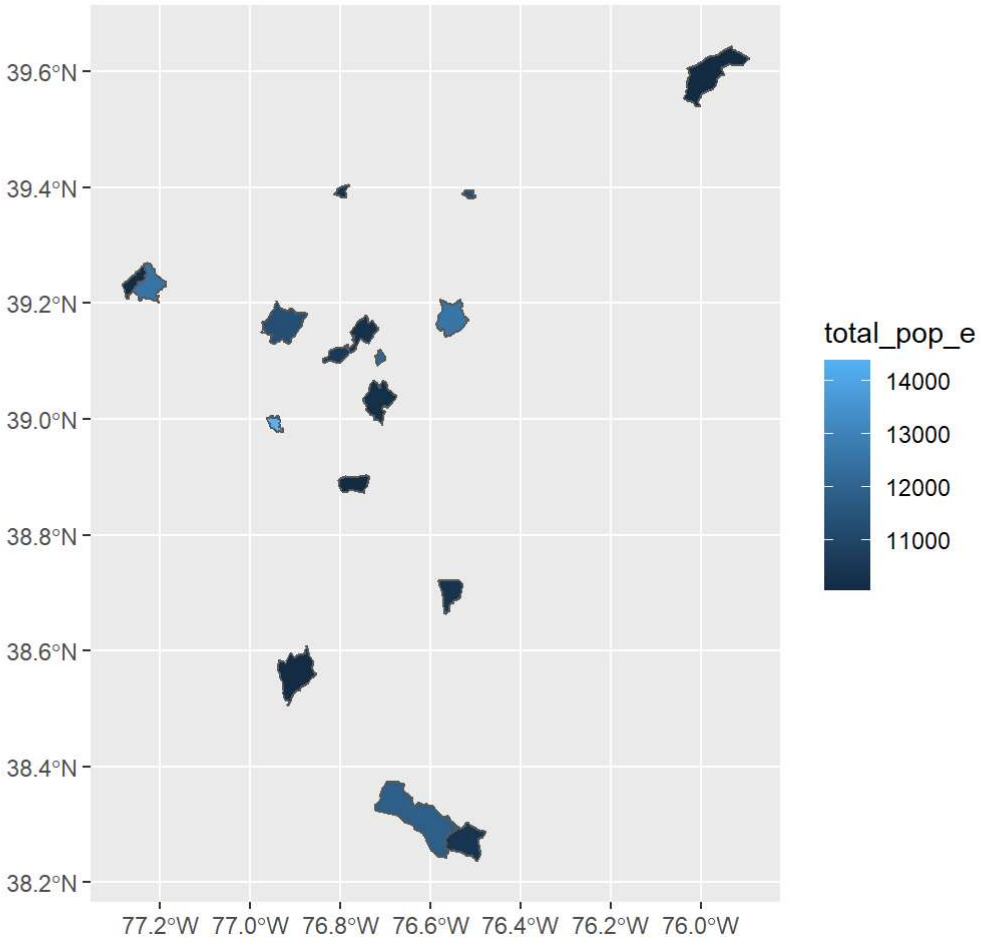



```
#Filtered Map (Total Pop. > 10,000)
Maryland10k = filter(Maryland5year, total_pop_e>10000)

#Plotting Filtered Map
ggplot(Maryland10k) + geom_sf(aes(fill = total_pop_e))
```

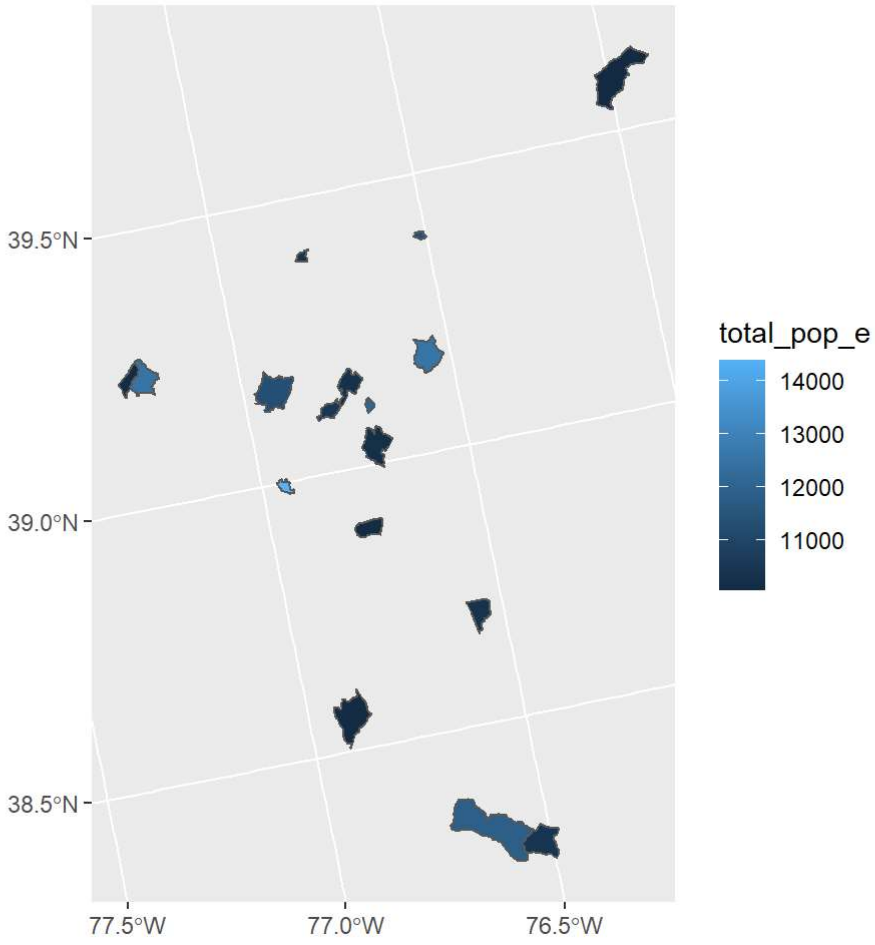


5. Reproject your census tract data to USA Contiguous Albers Equal Area Conic (http://spatialreference.org/ref/esri/usa-contiguous-albers-equal-area-conic/). Plot the original and the reprojected data sets. Do you see any difference? (3 points)

```
#Transforming Maryland Map

Marylandtransform = st_transform(Maryland5year, 5070)


Maryland10ktransform = st_transform(Maryland10k, 5070)



#plotting original and transformed datasets for Maryland map with Population above 10,000

ggplot(Maryland10k) + geom_sf(aes(fill = total_pop_e))
```




```
ggplot(Maryland10ktransform) + geom_sf(aes(fill = total_pop_e))
```




```
# After transforming the dataset, the census tracts themselves appear to be slightly smaller and elongated (almost un-noticeable), but the biggest change is regarding the plane that the census tracts are placed on. The grid itself is rotated counterclockwise as opposed to standing straight up like a traditional X,Y coordinate plane.
```

6. Create a new sf with only three counties in your state. What is the sum of the shape areas in these three counties? (3 points) Hint use `st_area()` .

```
#Loading County Datasets

Arundeldata2019 = get_acs(geography = "tract", year=2019, state = "MD", survey="acs5", county = "Anne Arundel",
                variables = c("Total Pop" = "B01003_001",
"Black Pop" = "B01001B_001",
                              "medHouseprice" = "B25077_001"),
                geometry = TRUE,
                output = "wide") %>% clean_names()
```

```
## Getting data from the 2015-2019 5-year ACS
```

```
## Downloading feature geometry from the Census website.  To cache shapefiles for use in future sessions, set `options(tigris_use_cache = TRUE)`.
```

```
Howarddata2019 = get_acs(geography = "tract", year=2019, state = "MD", survey="acs5", county =
"Howard",
                  variables = c("Total Pop" = "B01003_001",
"Black Pop" = "B01001B_001",
                                "medHouseprice" = "B25077_001"),
                  geometry = TRUE,
                  output = "wide") %>% clean_names()
```

```
## Getting data from the 2015-2019 5-year ACS
## Downloading feature geometry from the Census website.  To cache shapefiles for use in future
sessions, set `options(tigris_use_cache = TRUE)`.
```

```
Carrolldata2019 = get_acs(geography = "tract", year=2019, state = "MD", survey="acs5", county =
"Carroll",
                  variables = c("Total Pop" = "B01003_001",
"Black Pop" = "B01001B_001",
                                "medHouseprice" = "B25077_001"),
                  geometry = TRUE,
                  output = "wide") %>% clean_names()
```

```
## Getting data from the 2015-2019 5-year ACS
## Downloading feature geometry from the Census website.  To cache shapefiles for use in future
sessions, set `options(tigris_use_cache = TRUE)`.
```

```
mycounties = c(Carrolldata2019, Arundeldata2019, Howarddata2019)



carrollarea = st_area(Carrolldata2019)
arundlearea = st_area(Arundeldata2019)
howardarea = st_area(Howarddata2019)

countyarea = c(carrollarea, arundlearea, howardarea)


sum(countyarea)
```

```
## 2993025140 [m^2]
```

```
class(mycounties)
```

```
## [1] "list"
```

```
# The sum of the areas of the 3 selected counties come out to be 3,491,958,547 sq ft



#binding counties into data.frame
counties = rbind(Carrolldata2019, Howarddata2019, Arundeldata2019)


counties
```

```
## Simple feature collection with 198 features and 8 fields (with 1 geometry empty)
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -77.31151 ymin: 38.71274 xmax: -76.3945 ymax: 39.7208
## geographic CRS: NAD83
## First 10 features:
##          geoid                                             name total_pop_e
## 1  24013507601 Census Tract 5076.01, Carroll County, Maryland        4609
## 2  24013509002 Census Tract 5090.02, Carroll County, Maryland        2841
## 3  24013507801 Census Tract 5078.01, Carroll County, Maryland        6222
## 4  24013507802 Census Tract 5078.02, Carroll County, Maryland        4736
## 5  24013507703 Census Tract 5077.03, Carroll County, Maryland        3038
## 6  24013513001 Census Tract 5130.01, Carroll County, Maryland        4883
## 7  24013505207 Census Tract 5052.07, Carroll County, Maryland        3875
## 8  24013504202 Census Tract 5042.02, Carroll County, Maryland        6581
## 9  24013514201 Census Tract 5142.01, Carroll County, Maryland        5054
## 10 24013505102 Census Tract 5051.02, Carroll County, Maryland        5709
##    total_pop_m black_pop_e black_pop_m med_houseprice_e med_houseprice_m
## 1          382         181          92           309600            24788
## 2          252          82          67           393300            23622
## 3          578         487         166           272100            44696
## 4          402          77          66           309500            24848
## 5          208         161         130           293000            36472
## 6          299         182          96           394400            22106
## 7          215         112          82           342800            28872
## 8          400          88          80           384000            21679
## 9          206         135         122           403300            21987
## 10         221          43          53           385200            14012
##                          geometry
## 1  MULTIPOLYGON (((-77.0061 39...
## 2  MULTIPOLYGON (((-77.13345 3...
## 3  MULTIPOLYGON (((-77.01734 3...
## 4  MULTIPOLYGON (((-77.03848 3...
## 5  MULTIPOLYGON (((-77.06041 3...
## 6  MULTIPOLYGON (((-77.16738 3...
## 7  MULTIPOLYGON (((-76.97923 3...
## 8  MULTIPOLYGON (((-77.00533 3...
## 9  MULTIPOLYGON (((-77.09008 3...
## 10 MULTIPOLYGON (((-76.95053 3...
```

```
class(counties)
```

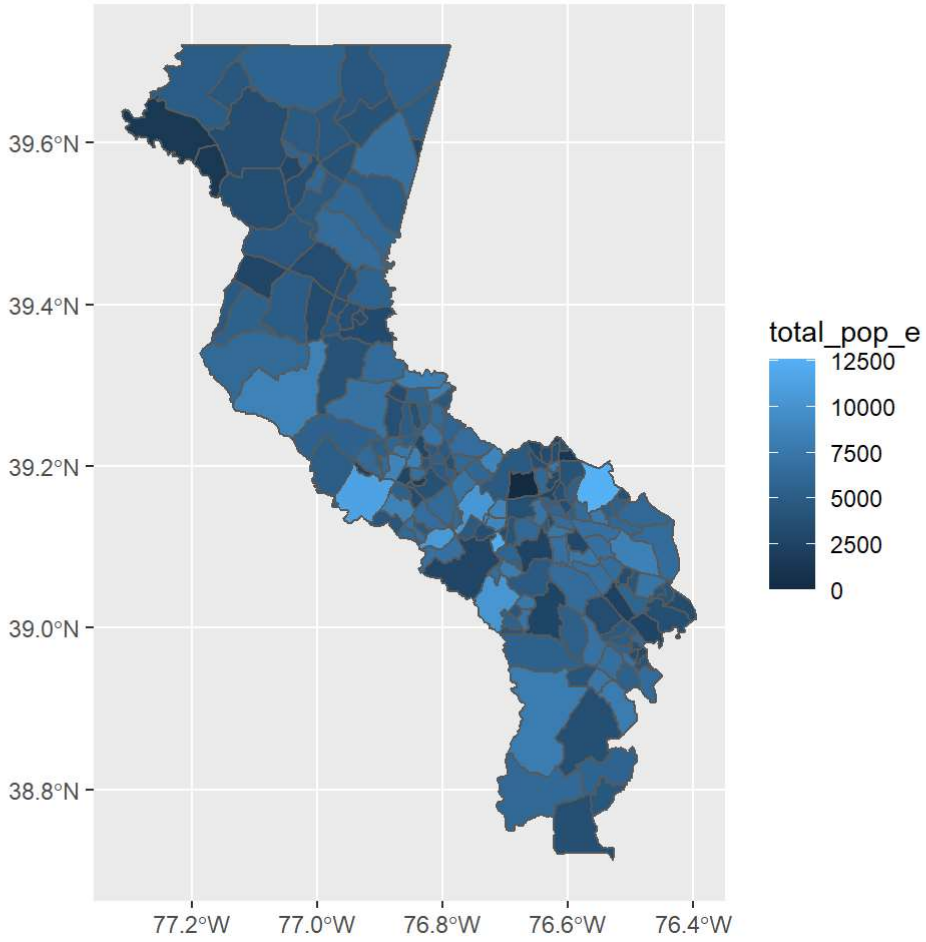
```
## [1] "sf"         "data.frame"
```

7. Plot the new sf you created in question 6 using ggplot2.

```
#Grouping Selected Counties


countygroups <- counties %>% group_by(total_pop_e) %>% summarise(counties)
```

```
## `summarise()` has grouped output by 'total_pop_e'. You can override using the `.groups` argum
ent.
```

```
ggplot(countygroups) + geom_sf(aes(fill = total_pop_e))
```

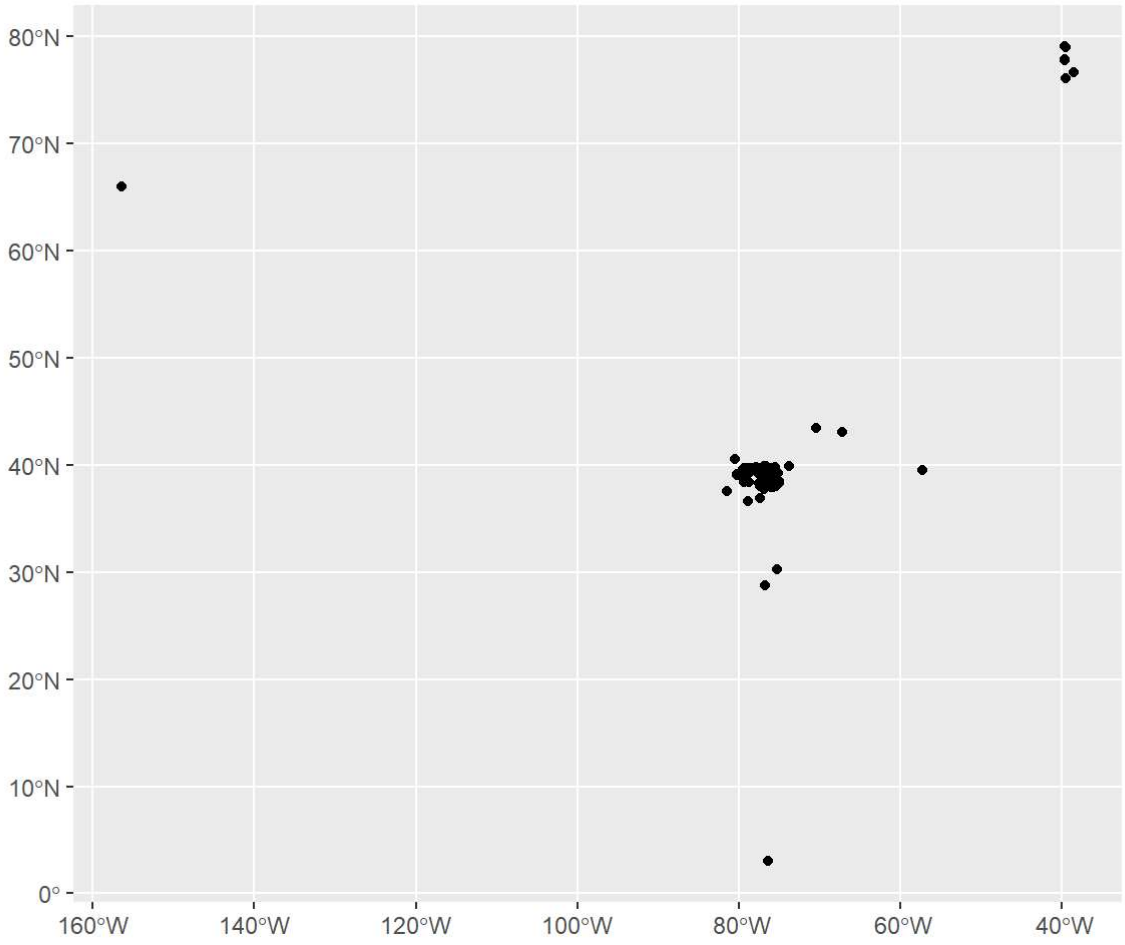


8. The Environmental Protection Agency keeps a Toxics Release Inventory (TRI) of toxic chemical releases and pollution prevention activities reported by industrial and federal facilities. Download the TRI data for the state you are mapping in this assignment: https://www.epa.gov/frs/epa-frs-facilities-state-single-file-csv-download (https://www.epa.gov/frs/epa-frs-facilities-state-single-file-csv-download). Read your TRI .csv into R and turn it into a MULTIPOINT spatial object. (2 points)

```
# Loading in CSV
TRI = read.csv("C:/Users/justi/Downloads/School/GES 687/MD_FACILITY_FILE.CSV", stringsAsFactors
 = FALSE)

colnames(TRI)
```

```
##  [1] "FRS_FACILITY_DETAIL_REPORT_URL" "REGISTRY_ID"
##  [3] "PRIMARY_NAME"                   "LOCATION_ADDRESS"
##  [5] "SUPPLEMENTAL_LOCATION"          "CITY_NAME"
##  [7] "COUNTY_NAME"                    "FIPS_CODE"
##  [9] "STATE_CODE"                     "STATE_NAME"
## [11] "COUNTRY_NAME"                   "POSTAL_CODE"
## [13] "FEDERAL_FACILITY_CODE"          "FEDERAL_AGENCY_NAME"
## [15] "TRIBAL_LAND_CODE"               "TRIBAL_LAND_NAME"
## [17] "CONGRESSIONAL_DIST_NUM"         "CENSUS_BLOCK_CODE"
## [19] "HUC_CODE"                       "EPA_REGION_CODE"
## [21] "SITE_TYPE_NAME"                 "LOCATION_DESCRIPTION"
## [23] "CREATE_DATE"                    "UPDATE_DATE"
## [25] "US_MEXICO_BORDER_IND"           "PGM_SYS_ACRNMS"
## [27] "LATITUDE83"                     "LONGITUDE83"
## [29] "CONVEYOR"                       "COLLECT_DESC"
## [31] "ACCURACY_VALUE"                 "REF_POINT_DESC"
## [33] "HDATUM_DESC"                    "SOURCE_DESC"
```

```
TRIfilte = filter(TRI, LATITUDE83>0)

#Transforming CSV to multi-point object
TRImultipoint <- st_as_sf(TRIfilte, coords = c("LONGITUDE83", "LATITUDE83"), crs = 4269)

ggplot(TRImultipoint) + geom_sf()
```



9. Plot TRI data in only the 3 counties you selected in question 6. (3 points)

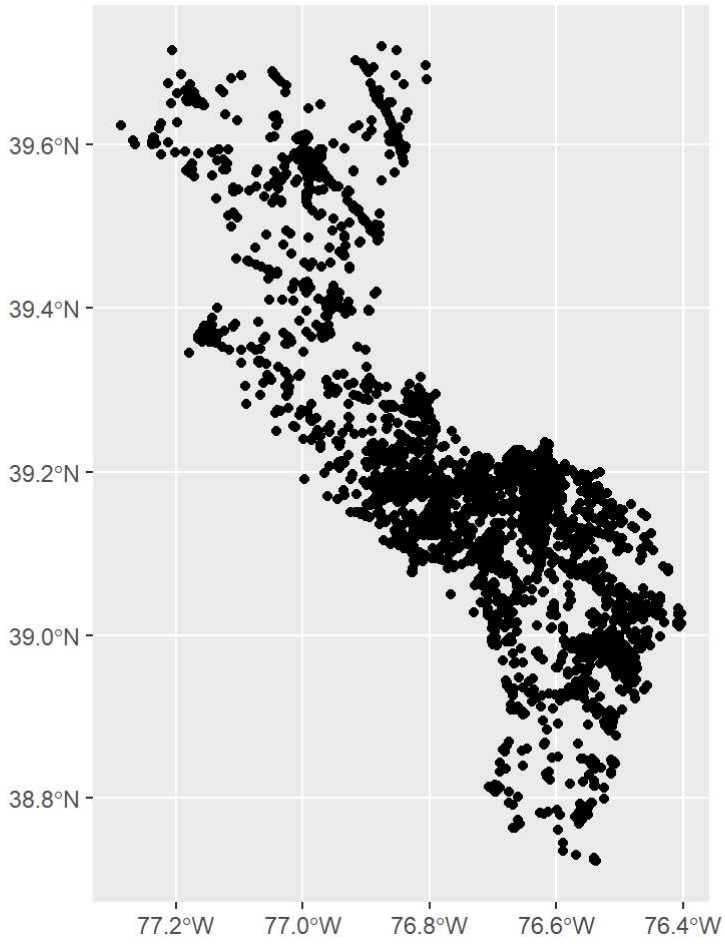
```
TRItransform = st_transform(TRImultipoint, 3857)

countygroupstransform = st_transform(countygroups, 3857)

st_crs(TRItransform) == st_crs(countygroupstransform)
```

```
## [1] TRUE
```

```
TRIcounty = st_intersection(countygroupstransform, TRItransform)
```

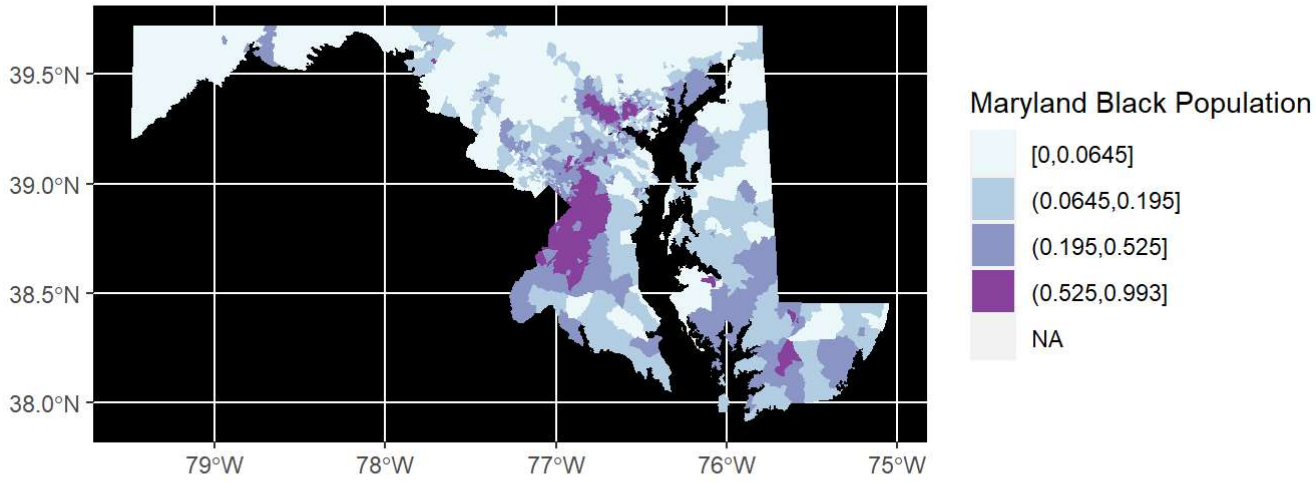
```
## Warning: attribute variables are assumed to be spatially constant throughout all
## geometries
```

```
ggplot(TRIcounty) + geom_sf(aes())
```



10. Use ggplot2 to create a choropleth map of your state for the second Census variable you selected in question 1.

```
Maryland5year$blackprop = Maryland5year$black_pop_e/Maryland5year$total_pop_e


Maryland5year %>%
 ggplot() +
 geom_sf(aes(fill =
 cut_number(blackprop, 4)), color = NA) +
 scale_fill_brewer(palette = "BuPu", "name" = "Maryland Black Population") +
 theme(panel.background = element_rect(fill = "black", color = "black"))
```



Part 3. Reflection (3 points)

```
#This assignment was very useful for teaching me more about how to perform spatial analysis in
 R. I was able to deepen my skills with using various st_ commands to transform coordinates and
 edit map data, as well as gaining more practice with plotting colorful and informative maps.

#The most important skill I think I learned from this assignment was how to more accurately trou
bleshoot data, helping me figure out problems without needing help. I specifically had a bit of
 trouble with uploading the TRI data and intersecting the TRI data with my county maps, but lear
ning how to edit CSV files in excel to better develop my maps in R greatly helped in my own proj
ect which requires me to constantly move files between excel, R, QGIS, and STATA. I also learned
a lot more about how to edit maps using ggplot and feel more comfortable with editing maps witho
ut just copying and pasting code.
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.