Project 3. Thematic Mapping and Data Classification

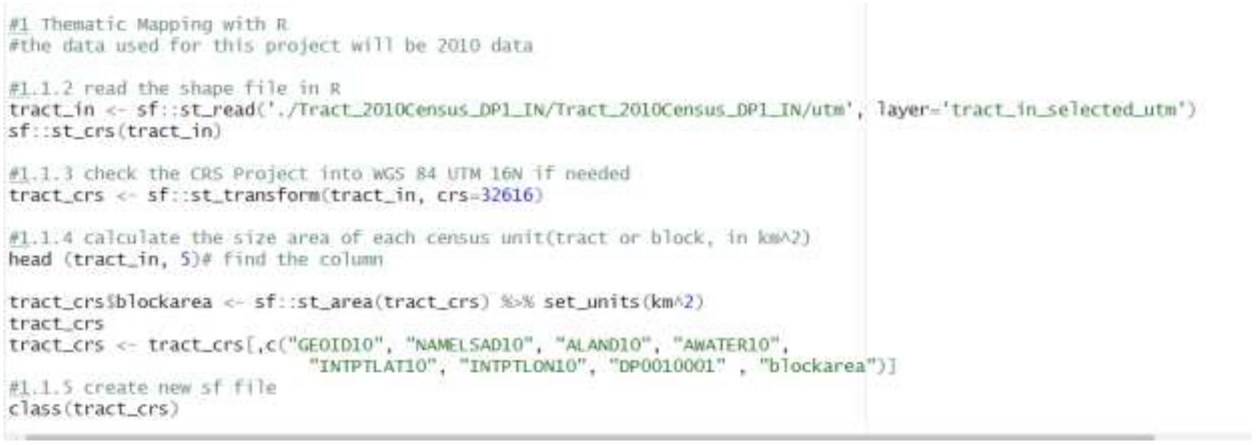
Abstract

The purpose of this project is to practice downloading Census Tract data, convert them into shapefiles, and conduct population density analysis. The project analyzed the population density based on four classification methods: Equal Interval, Quantile, Standard Deviation, and Jenks. In addition, the project also practiced finding the break point of each classification method, analyze the pros and cons of four different classification methods. At last, the project compared the tract and county population density.

1. Thematic Mapping with R (50 pts)

1.1 Data Preparation

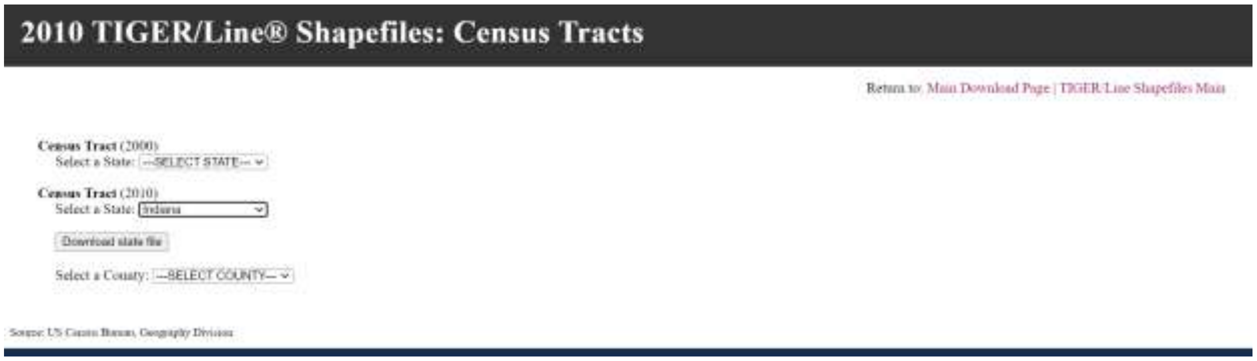
The first part of the project is to either find the Census data and implement shapefiles. In addition, this part of the project will convert it to WGS 84 UTM 16 N. Additionally, this section calculated the size area of the Census unit, which is the tract unit, into square kilometers. The related code is show below.



```
#1 Thematic Mapping with R
#the data used for this project will be 2010 data

#1.1.2 read the shape file in R
tract_in <- sf::st_read('./Tract_2010Census_DP1_IN/Tract_2010Census_DP1_IN/utm', layer='tract_in_selected_utm')
sf::st_crs(tract_in)

#1.1.3 check the CRS Project into WGS 84 UTM 16N if needed
tract_crs <- sf::st_transform(tract_in, crs=32616)

#1.1.4 calculate the size area of each census unit(tract or block, in km^2)
head (tract_in, 5)# find the column

tract_crs$blockarea <- sf::st_area(tract_crs) %>% set_units(km^2)
tract_crs
tract_crs <- tract_crs[,c("GEOID10", "NAMELSAD10", "ALAND10", "AWATER10",
                          "INTPTLAT10", "INTPTLON10", "DP0010001" , "blockarea")]
#1.1.5 create new sf file
class(tract_crs)
```

The new converted unit area is listed as “blockarea” in a new column.

During this part of the project, I used the sample data provided by the instructor. However, to download the files from Census, try to go to the Census website and find the 2010 TIGER/Line Shapefiles and download Census Tract data for the state of Indiana. The following screenshot shows the place of downloading the data corresponding data.



2010 TIGER/Line® Shapefiles: Census Tracts

Return to: Main Download Page | TIGER/Line Shapefiles Main

Census Tract (2000)
Select a State: ---SELECT STATE---

Census Tract (2010)
Select a State: Indiana

Select a County: ---SELECT COUNTY---

Source: US Census Bureau, Geography Division

In section 3, as there is no data for county from the instructor's sample files, it is mandatory to get to the Census website and download the county data. More details will be discussed in section 3.

1.2 Map Design

This part of the project is to draw a map layout that can be produce a as commercial or professional tools. After calculating the population density, boarder lines, scale bar, north pointer, lagend and data source has been implemented into the map. In addition, the map also included the date of creation. Figure 01 shows the sample of the density population map.

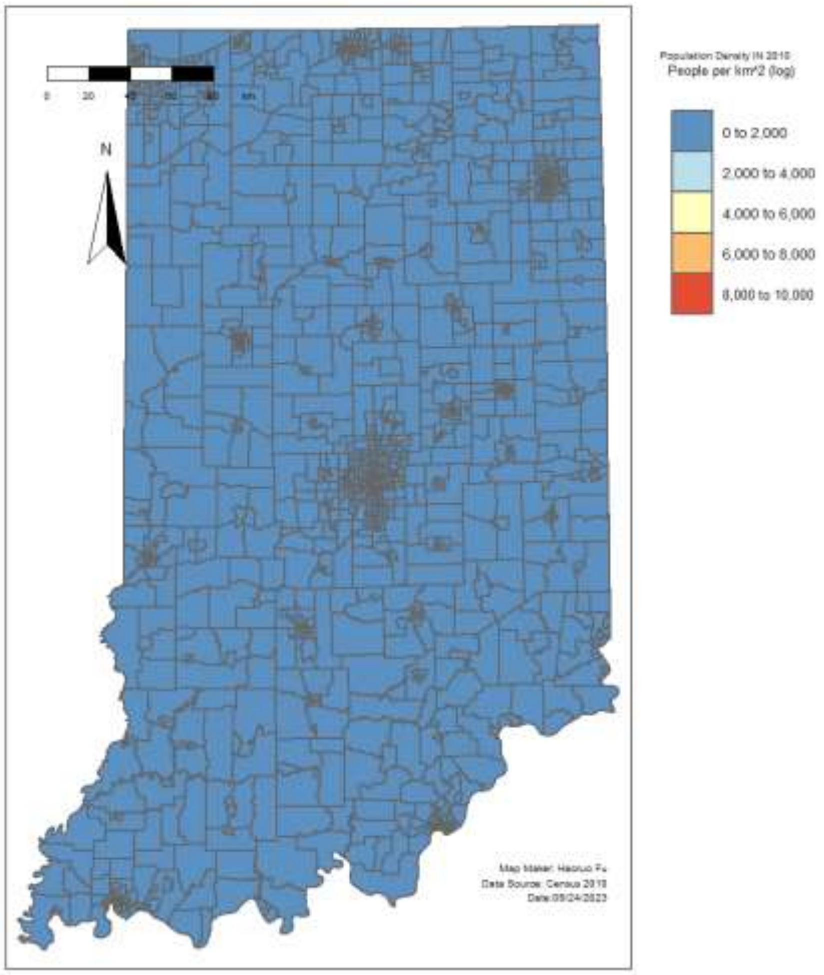


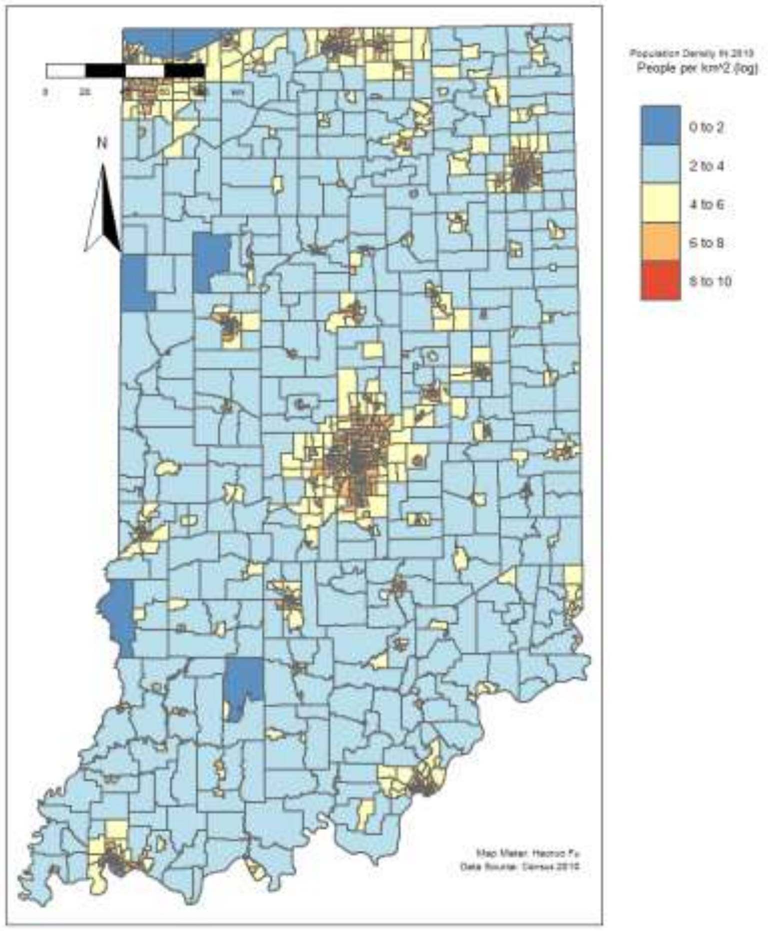
Figure 01

Density Population (1.2)

From the map, we can see that most of the tract block has the population of less than 2,000 per square kilometers. Thus, I tried to use population density log to have better visualization for references. Figure 02 shows an example of the population density in log version. In the following part of the project, the additional log version of the map will be listed in the reference. The code for this part of the project is also listed in reference.

Figure 02

Density Population (log)

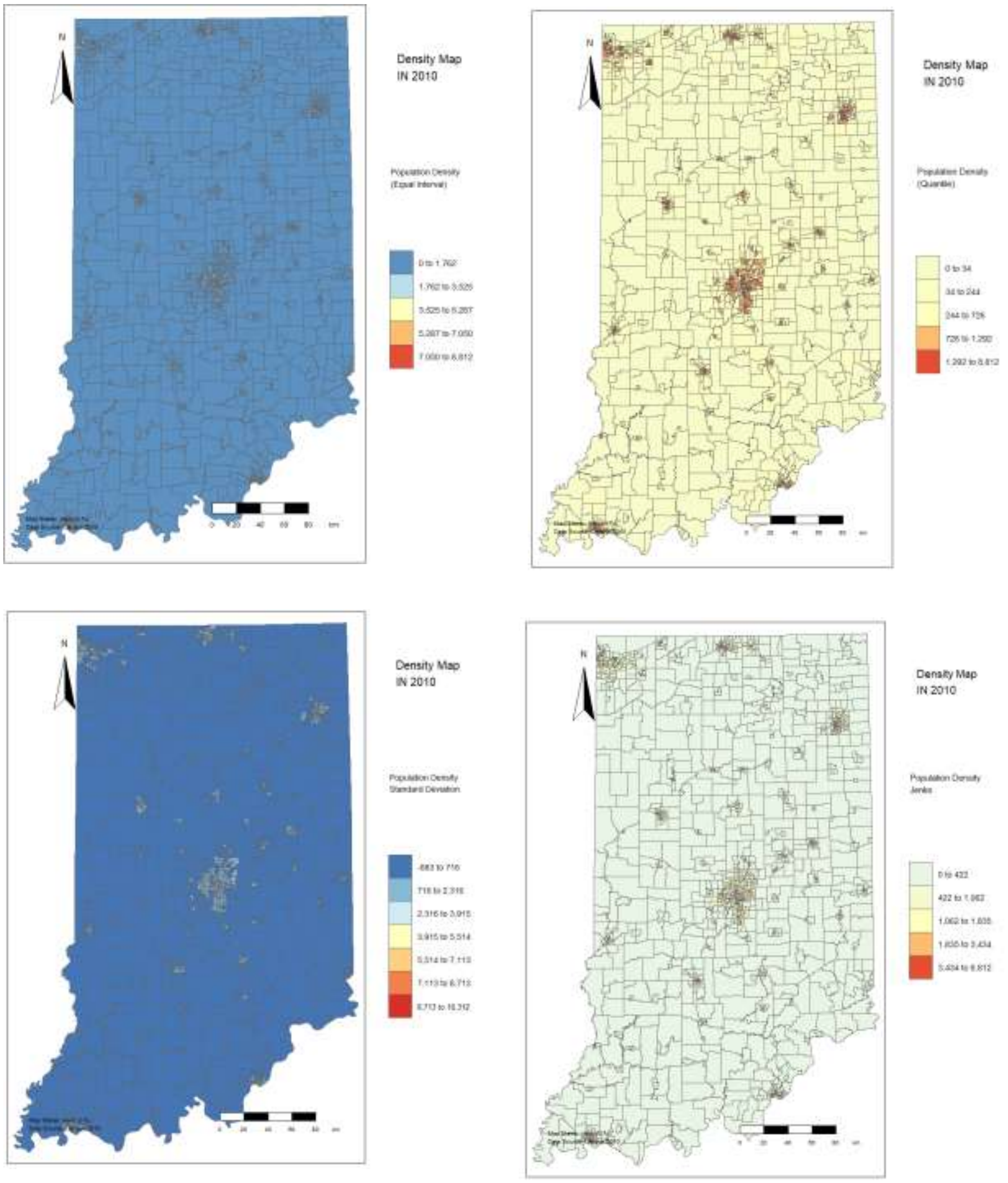


1.3 Classification Method Evaluation

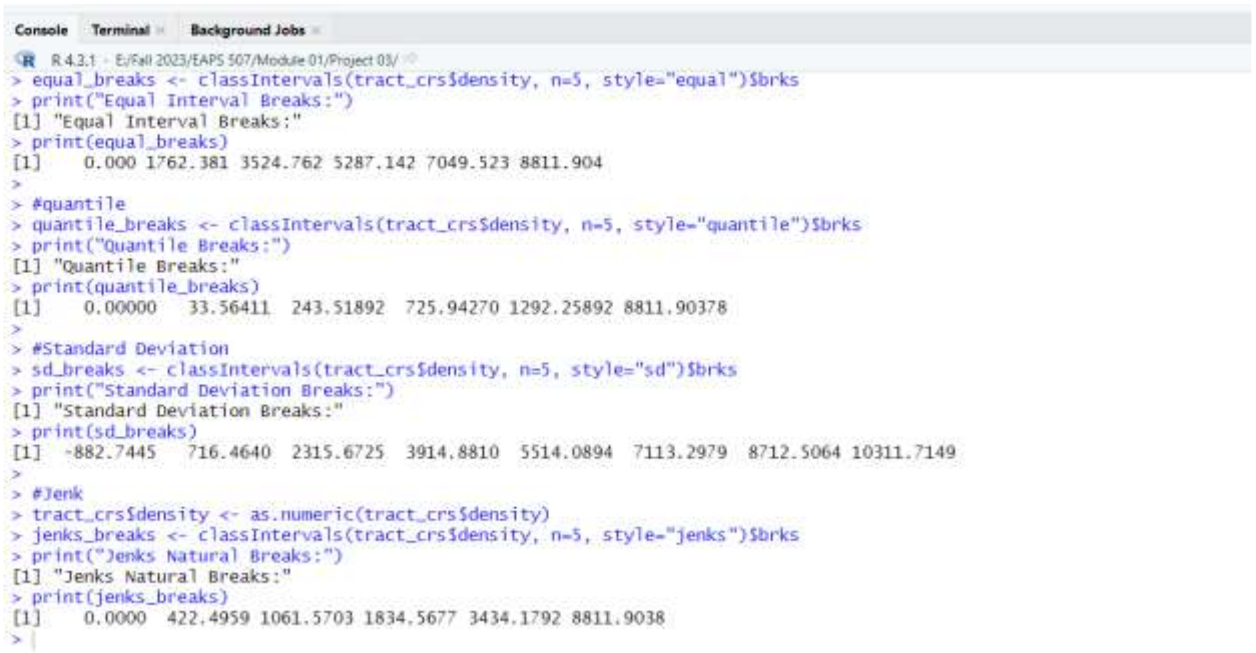
This part of the project used four R built-in classification methods to produce population density maps. In this section, Equal Interval, Quantile, Standard Deviation, and Jenks were used to draw the density maps. In addition, this section finds out the break point of each classification method. Figure 03 shows mapping of four different classification methods.

Figure 03

Equal Interval, Quantile, SD, and Jenks Classification Density Maps



After the mapping, the breaks of each classification method has been printed out in the following screenshot. For equal interval, quantile, and Jenks, it all limited to 5 breaks (six numbers, or points, showing) and for the Standard Deviation, the code placed it into 7 breaks (8 numbers).



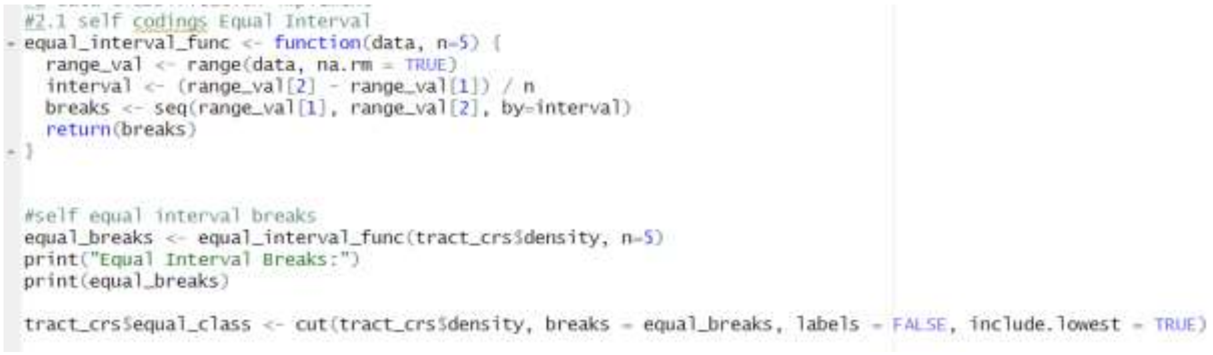
Console Terminal Background Jobs

R 4.3.1 · E:/Fall 2023/EAPS 507/Module 01/Project 03/

```
> equal_breaks <- classIntervals(tract_crs$density, n=5, style="equal")$brks
> print("Equal Interval Breaks:")
[1] "Equal Interval Breaks:"
> print(equal_breaks)
[1]    0.000 1762.381 3524.762 5287.142 7049.523 8811.904
>
> #quantile
> quantile_breaks <- classIntervals(tract_crs$density, n=5, style="quantile")$brks
> print("Quantile Breaks:")
[1] "Quantile Breaks:"
> print(quantile_breaks)
[1]    0.00000   33.56411  243.51892  725.94270 1292.25892 8811.90378
>
> #Standard Deviation
> sd_breaks <- classIntervals(tract_crs$density, n=5, style="sd")$brks
> print("Standard Deviation Breaks:")
[1] "Standard Deviation Breaks:"
> print(sd_breaks)
[1]  -882.7445   716.4640  2315.6725  3914.8810  5514.0894  7113.2979  8712.5064 10311.7149
>
> #Jenk
> tract_crs$density <- as.numeric(tract_crs$density)
> jenks_breaks <- classIntervals(tract_crs$density, n=5, style="jenks")$brks
> print("Jenks Natural Breaks:")
[1] "Jenks Natural Breaks:"
> print(jenks_breaks)
[1]    0.0000  422.4959 1061.5703 1834.5677 3434.1792 8811.9038
> 
```

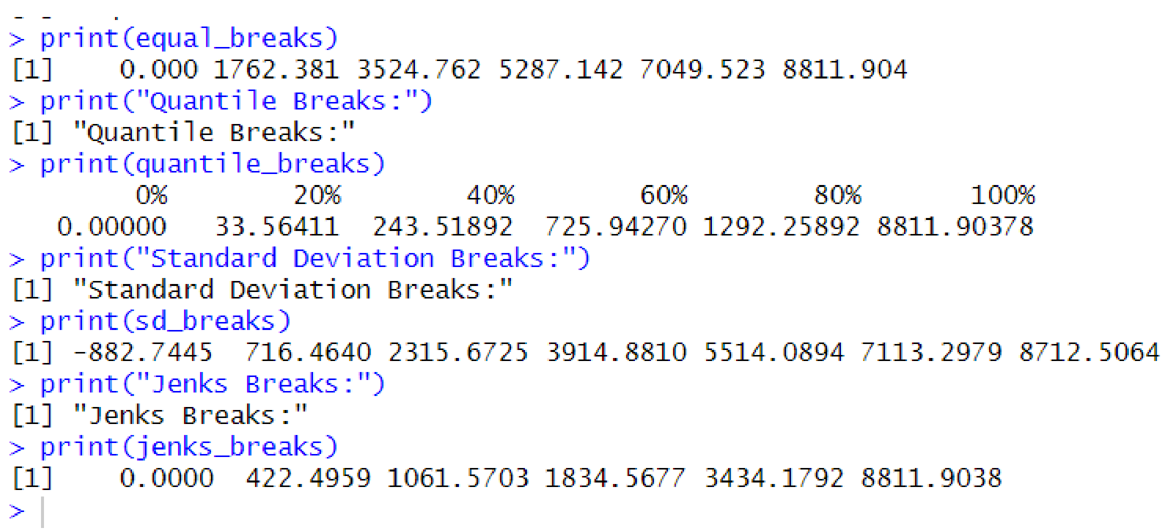
2. Data Classification Implementation (40 pts)

This section asks to code by myself of the four classification method. The following screenshot shows an example of the equal interval coding. The rest of the coding is listed in the reference. The related map is also showing in the reference.



```
#2.1 self codings Equal Interval
equal_interval_func <- function(data, n=5) {
  range_val <- range(data, na.rm = TRUE)
  interval <- (range_val[2] - range_val[1]) / n
  breaks <- seq(range_val[1], range_val[2], by=interval)
  return(breaks)
}


#self equal interval breaks
equal_breaks <- equal_interval_func(tract_crs$density, n=5)
print("Equal Interval Breaks:")
print(equal_breaks)

tract_crs$equal_class <- cut(tract_crs$density, breaks = equal_breaks, labels = FALSE, include.lowest = TRUE)
```

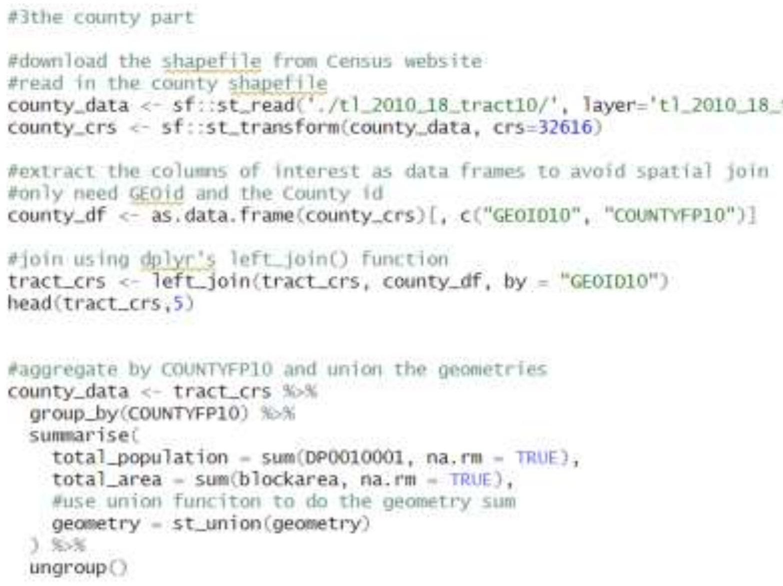
For the breaks, equal interval, quantile are having the same break numbers. However, it is necessary to modify the standard deviation in order to receive the same result. I have modified the sequence starts from two standard deviations below the mean and extends to ten value above the mean, by every two values, in order to get the same result from the default R built-in classification. The following screenshot shows the break point of my own code.



```
> print(equal_breaks)
[1]    0.000 1762.381 3524.762 5287.142 7049.523 8811.904
> print("Quantile Breaks:")
[1] "Quantile Breaks:"
> print(quantile_breaks)
        0%        20%        40%        60%        80%       100%
   0.00000   33.56411  243.51892  725.94270 1292.25892 8811.90378
> print("Standard Deviation Breaks:")
[1] "Standard Deviation Breaks:"
> print(sd_breaks)
[1] -882.7445  716.4640 2315.6725 3914.8810 5514.0894 7113.2979 8712.5064
> print("Jenks Breaks:")
[1] "Jenks Breaks:"
> print(jenks_breaks)
[1]    0.0000  422.4959 1061.5703 1834.5677 3434.1792 8811.9038
> 
```

3. **Examine the MAUP phenomenon (20 pts)**

   This part of the project will shift into county and do the same as above. The original data provided by instructor does not have the county column (COUNTYFP10). Thus, in order to do this section, we need to download the county shapefiles first. After downloading the county shapefiles from the Census, we can see that the county shapefiles does not have any data related to population. Thus, we need to add the county column to the original "tract_crs" file we used in the previous section. We first add the "COUNTYFP10" column to the "tract_crs" file based on "GEOID10" column, as they are the same. Then, we would need to group the "blockarea (tract)" to the new county area using the "st_union" function. After this, we are able to draw the county population density map. The following screenshot provides a sample code for how to add a county column to the original tract file.

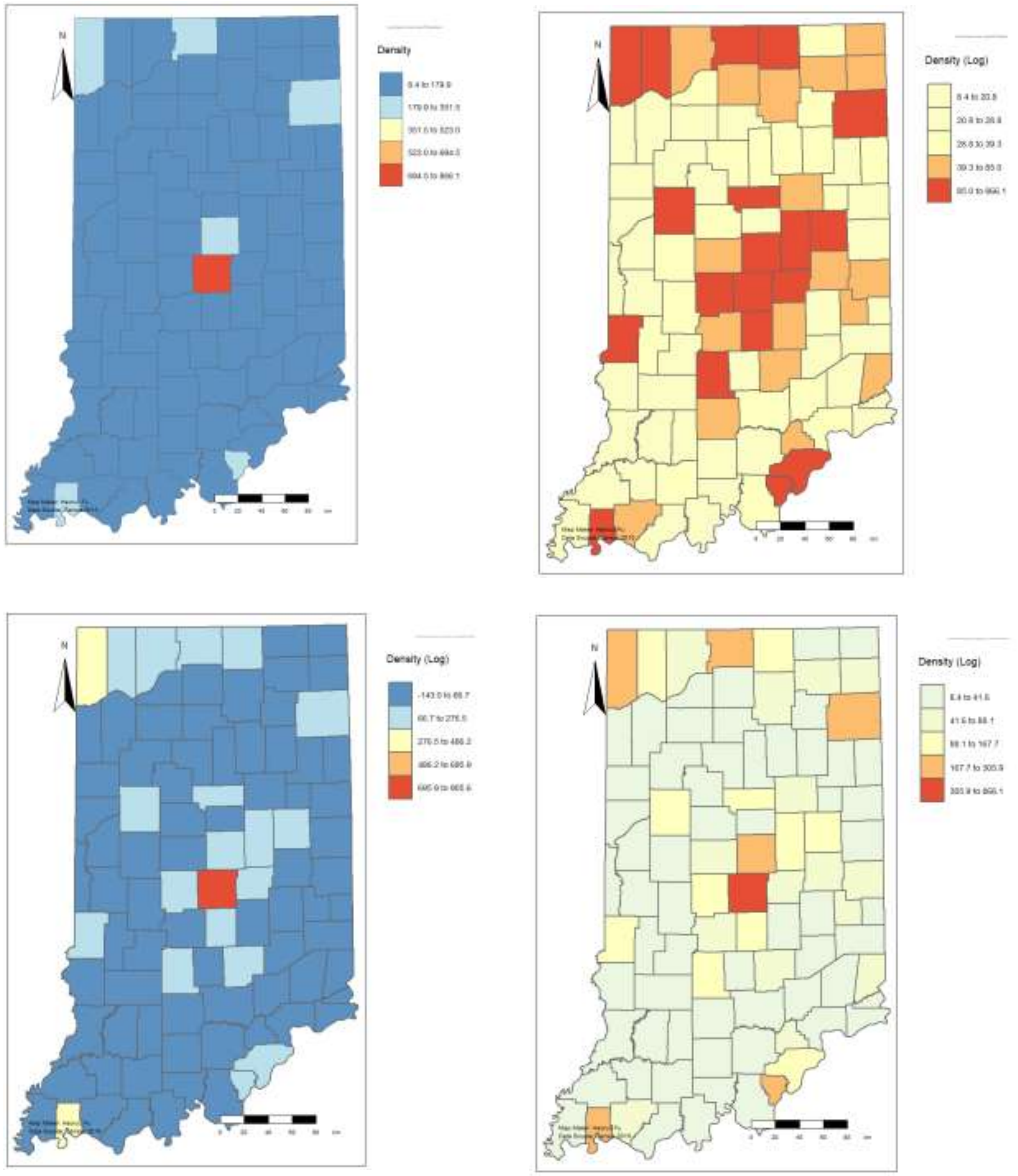


```
#3the county part

#download the shapefile from Census website
#read in the county shapefile
county_data <- sf::st_read('./tl_2010_18_tract10/', layer='tl_2010_18_tract10')
county_crs <- sf::st_transform(county_data, crs=32616)

#extract the columns of interest as data frames to avoid spatial join
#only need GEOid and the County id
county_df <- as.data.frame(county_crs)[, c("GEOID10", "COUNTYFP10")]

#join using dplyr's left_join() function
tract_crs <- left_join(tract_crs, county_df, by = "GEOID10")
head(tract_crs,5)


#aggregate by COUNTYFP10 and union the geometries
county_data <- tract_crs %>%
  group_by(COUNTYFP10) %>%
  summarise(
    total_population = sum(DP0010001, na.rm = TRUE),
    total_area = sum(blockarea, na.rm = TRUE),
    #use union funciton to do the geometry sum
    geometry = st_union(geometry)
  ) %>%
  ungroup()
```

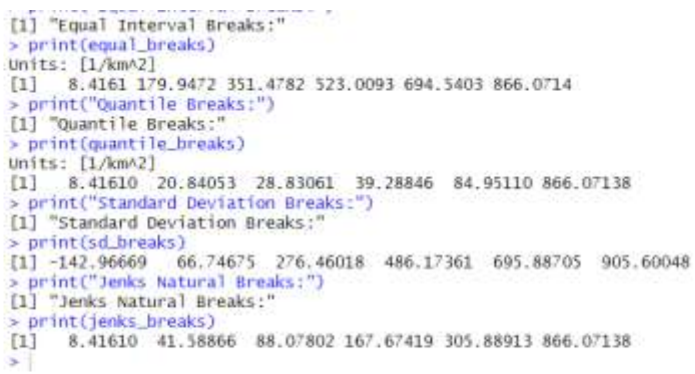
After the data preparation, I drew the density map with four classification methods. The following graphs shows the population density based on four classification methods (Figure 04).

Figure 04

Equal Interval, Quantile, SD, Jenks Density Map (County Level)



The break points of the county part is also showing in the following screenshot. All the classification method were break into 5 classes (6 number shown).



```
[1] "Equal Interval Breaks:"
> print(equal_breaks)
Units: [1/km^2]
[1]   8.4161 179.9472 351.4782 523.0093 694.5403 866.0714
> print("Quantile Breaks:")
[1] "Quantile Breaks:"
> print(quantile_breaks)
Units: [1/km^2]
[1]   8.41610  20.84053  28.83061  39.28846  84.95110 866.07138
> print("Standard Deviation Breaks:")
[1] "Standard Deviation Breaks:"
> print(sd_breaks)
[1] -142.96669   66.74675  276.46018  486.17361  695.88705  905.60048
> print("Jenks Natural Breaks:")
[1] "Jenks Natural Breaks:"
> print(jenks_breaks)
[1]   8.41610  41.58866  88.07802 167.67419 305.88913 866.07138
> 
```

MAUP Phenomenon

The Modifiable Areal Unit Problem (MAUP) refers to the phenomenon wherein spatial patterns in data can change based on the size and shape of the spatial units being analyzed. The MAUP shows that while doing the spatial statistics, results can be different due to the scale (zoning) and aggregation (scale) at which analysis is performed.
By comparing two types of the map (Census Tract vs. County), the change of unit can result in different patterns of population density. Smaller unit Tract may be able to show high-density area, but that may get averaged-out when aggregate to larger area such as county. This can be referred to as the Scale Effect.

4. Evaluation and discussion (40 pts)

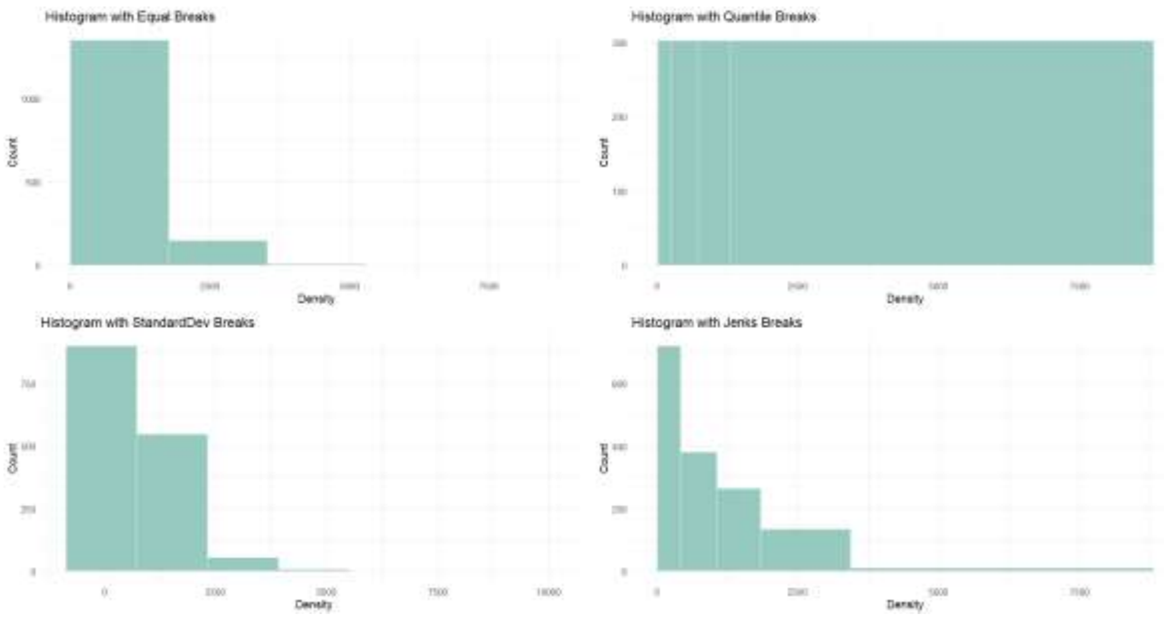
The following table indicates the break points of all four types of classifications.

Classification	Equal Interval	Quantile	SD	Jenks
Default R	0.000, 1762.381, 3254.762, 5287.142, 7049.523, 8811.904	0.000, 33.56411, 243.51892, 725.94270, 1292.25892, 8811.90378	-882.745, 716.4640, 2315.6725, 3914.8810, 5514.0894, 7113.2979, 8712.5064, 10311.7149	0, 422.4959, 1061.5703, 1834.5677, 3434.1792, 8811.9038
Self Code	0.000, 1762.381, 3254.762, 5287.142, 7049.523, 8811.904	0.000, 33.56411, 243.51892, 725.94270, 1292.25892, 8811.90378	-882.745, 716.4640, 2315.6725, 3914.8810, 5514.0894, 7113.2979, 8712.5064, 10311.7149	0, 0,0,0, 8811.904

The result showed that the default R code and the self code had the same breakpoint output. However, I would need to adjust the "sequence" function for the Standard Deviation part in order to get the correct break point. For the Jenks self code, I found that the Jenks manual code I did for myself showed the first and the last break point correct but the middle points are incorrect. One possible reason is that the when I was doing the sum of square variance matrix, it combined the previous variance and cause the issue.

Each classification method has its own pros and cons. For equal interval, it is easy to understand, but the data is not evenly distributed, some classes may have few counts while others may have many. If there are outliers, it may be hard to show the distribution well. For Quantile, each class contain the equal number of counts, however, it does not take actual data distribution into account. In other words, some values with very different amounts may be put into the same class. Standard Deviation emphasizes variation from the mean, but if the data is not normally distributed, it might be misleading. For the Jenks, it minimizes variance within classes and maximizes variance between classes, and it is useful for large dataset. In conclusion, Equal Interval may not be the best solution, since the population is hard to say regular distributed. Quantile may be a good one if we consider doing the ranking. Standard Deviation can be suggested only if it is normally distributed. Indiana density population is not a good project for using Standard Deviation. Jenks would be the best for this project analysis since a variable like population density, which might have "clusters" of similar values, this method will highlight the natural divisions between densely populated urban areas, less dense suburbs, and highly-populated rural areas. In addition, Jenks might be a good way to minimize the MAUP effect, because it determines the breaks based on natural groups in the data. But since it uses different levels (county and tract), it would be great to use smaller scale one (tract) rather than the county. In other words, if there is any larger scaled level (something larger than county), the result will be more general compared to tract level.

The following table shows histograms of four different classifications. We can see that the equal breaks show the problem of since it is not normally distributed, some class have a huge amount of data but the other classes may have few. The Quantile classification shows that each bin has the same amount, but we can see that the density range of one bin is larger than all other bins. Standard deviation seems to be a little better than the Equal Interval one, but still, they are having the same issue. Finally, it turns out that the Jenks would be the best classification method among these four methods.



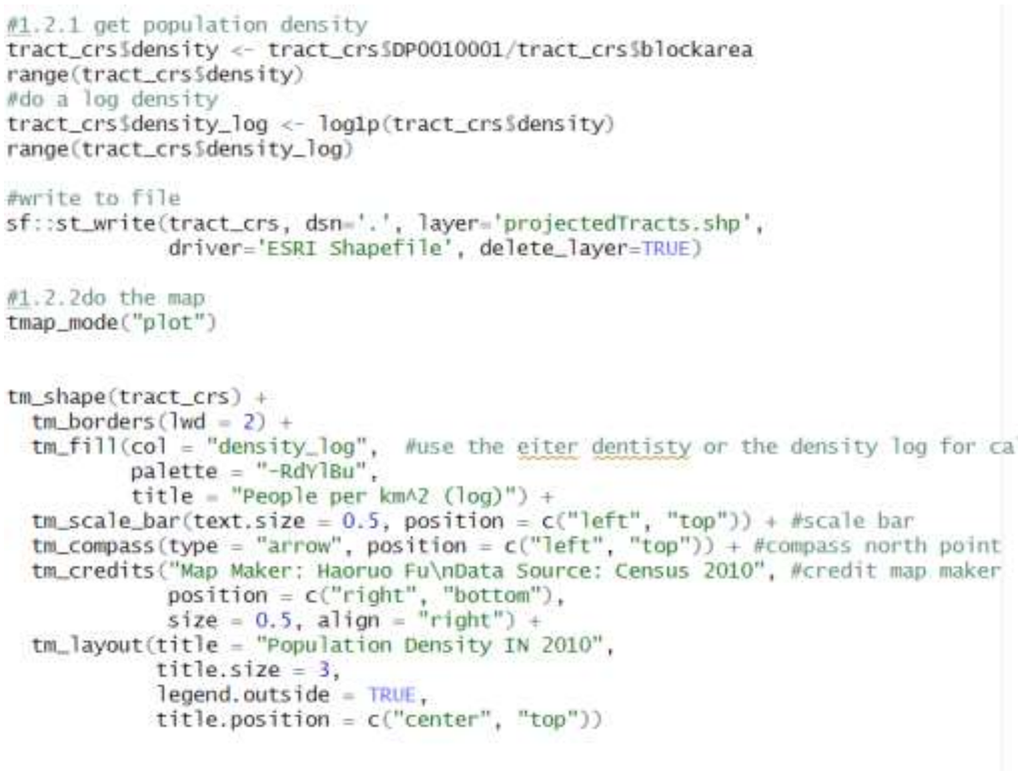
5. Summary/Conclusion/Concluding Remarks

This project conducts the population density analysis through four different types of classification methods and understand the usage (pros and cons) of four different types of method. In addition, this project also practiced how to access Census data, aggregate the shapefiles from different sources. While most of my projects were conducted based on the files provided by instructor, I found that accessing Census data is quite useful.

Acknowledgement

References

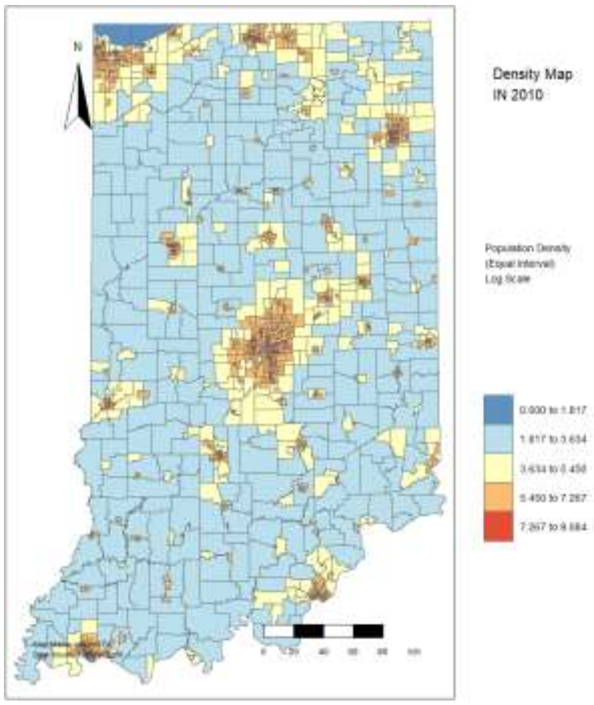
1.2 Coding part



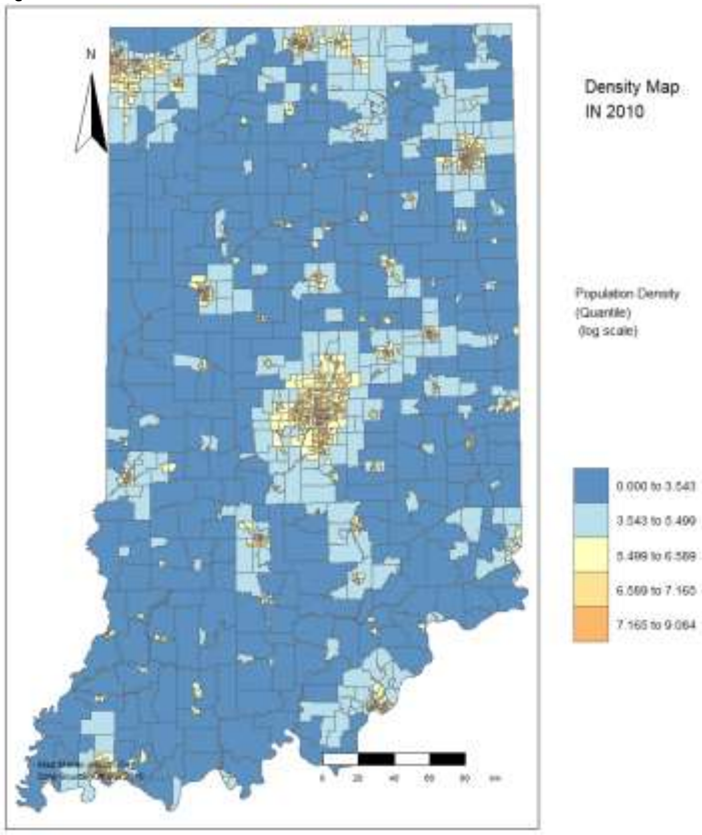
```
#1.2.1 get population density
tract_crs$density <- tract_crs$DP0010001/tract_crs$blockarea
range(tract_crs$density)
#do a log density
tract_crs$density_log <- log1p(tract_crs$density)
range(tract_crs$density_log)

#write to file
sf::st_write(tract_crs, dsn='.', layer='projectedTracts.shp',
             driver='ESRI Shapefile', delete_layer=TRUE)

#1.2.2do the map
tmap_mode("plot")


tm_shape(tract_crs) +
  tm_borders(lwd = 2) +
  tm_fill(col = "density_log",  #use the eiter dentisty or the density log for calculation
          palette = "-RdYlBu",
          title = "People per km^2 (log)") +
  tm_scale_bar(text.size = 0.5, position = c("left", "top")) + #scale bar
  tm_compass(type = "arrow", position = c("left", "top")) + #compass north point
  tm_credits("Map Maker: Haoruo Fu\nData Source: Census 2010", #credit map maker
             position = c("right", "bottom"),
             size = 0.5, align = "right") +
  tm_layout(title = "Population Density IN 2010",
            title.size = 3,
            legend.outside = TRUE,
            title.position = c("center", "top"))
```

1.4 Log Graphs

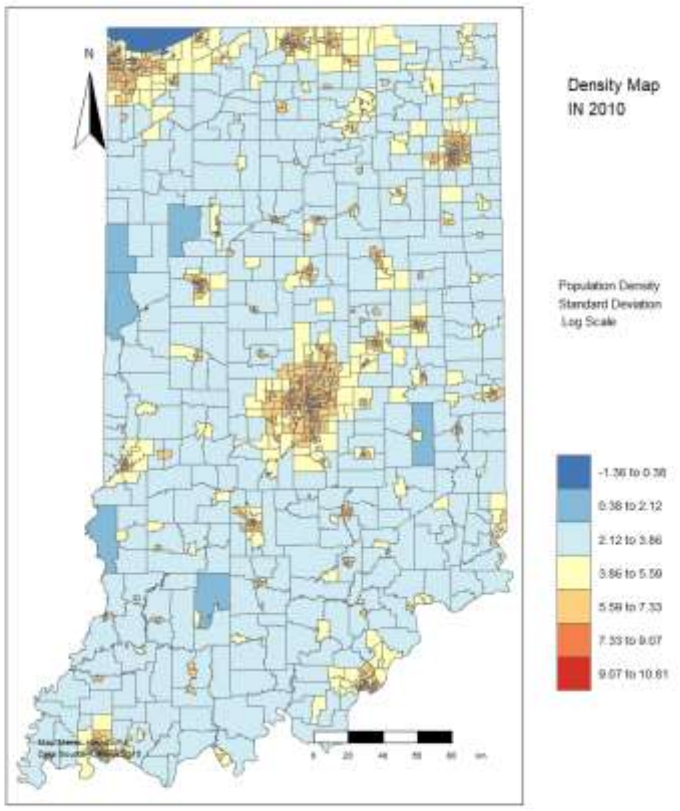
Equal Interval



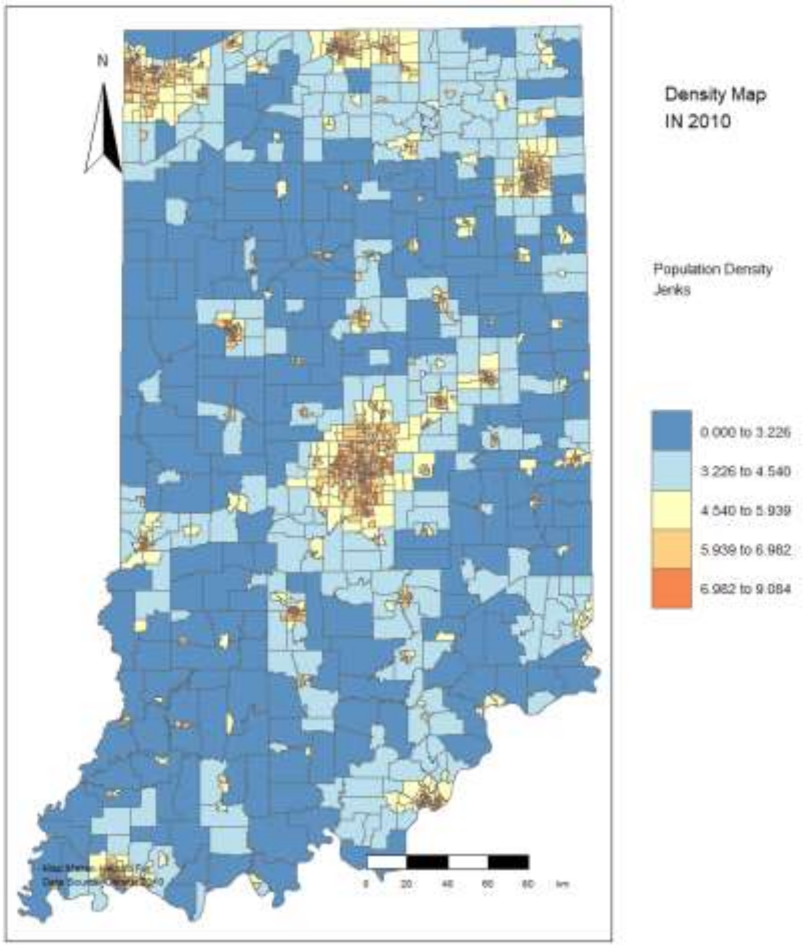
Quantile



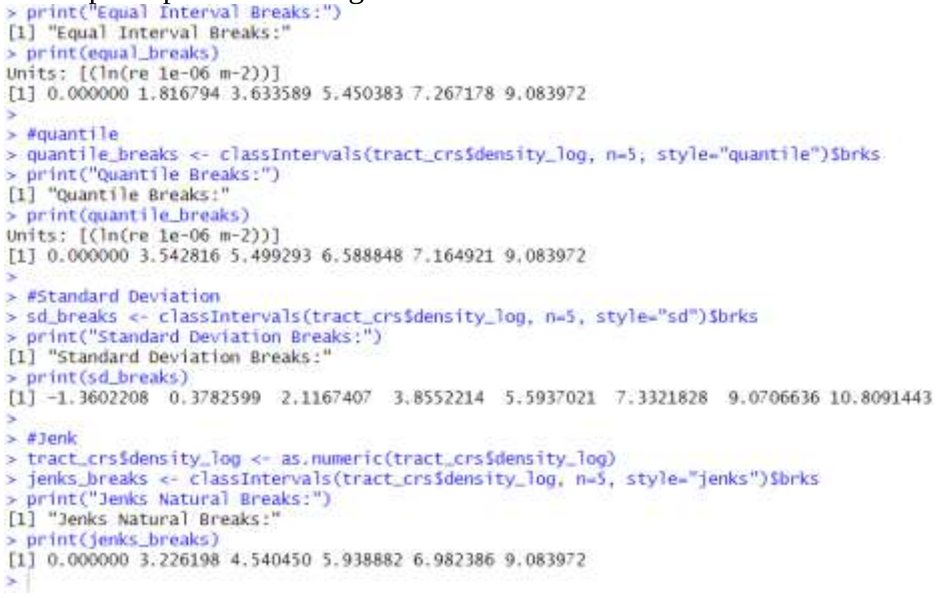
SD


N
Density Map
IN 2010
Population Density
Standard Deviation
Log Scale
-1.36 to 0.38
0.38 to 2.12
2.12 to 3.86
3.86 to 5.59
5.59 to 7.33
7.33 to 9.07
9.07 to 10.81
km

Jenks

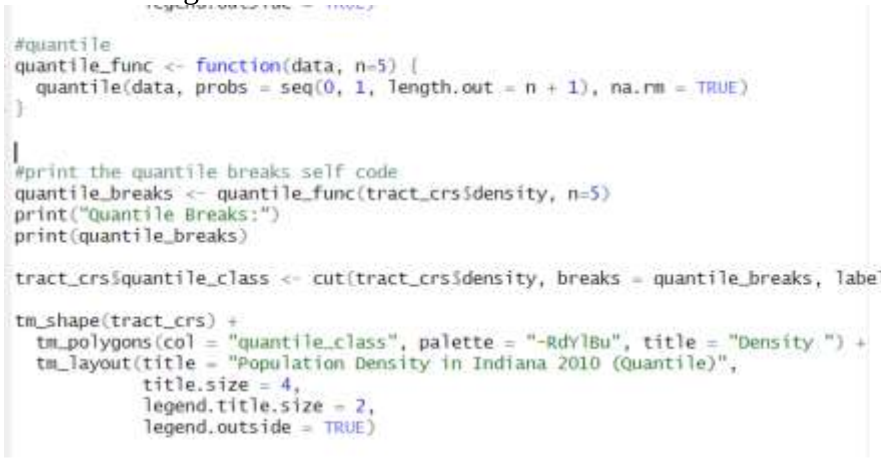


Break point print out in Log Scale

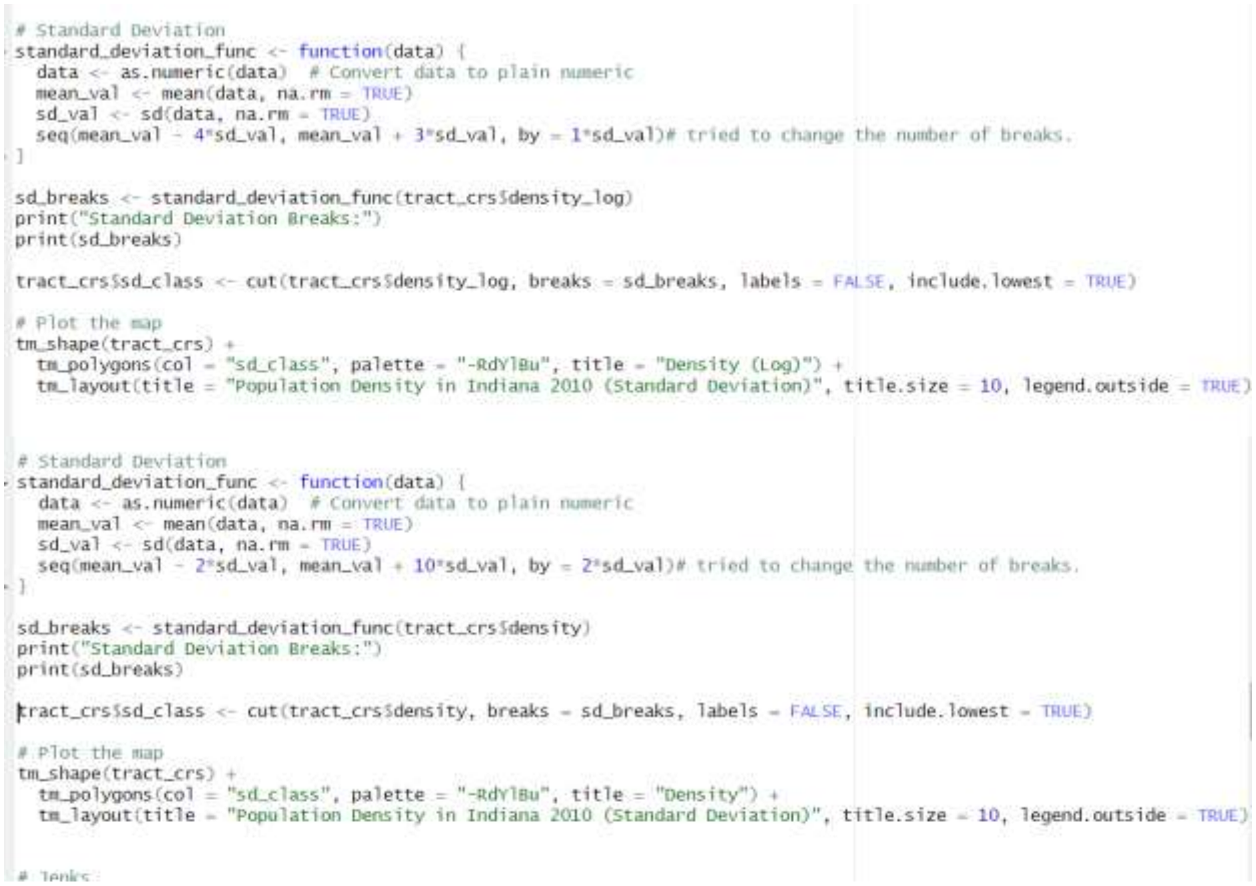


```
> print("Equal Interval Breaks:")
[1] "Equal Interval Breaks:"
> print(equal_breaks)
Units: [(ln(re 1e-06 m-2))]
[1] 0.000000 1.816794 3.633589 5.450383 7.267178 9.083972
>
> #quantile
> quantile_breaks <- classIntervals(tract_crs$density_log, n=5; style="quantile")$brks
> print("Quantile Breaks:")
[1] "Quantile Breaks:"
> print(quantile_breaks)
Units: [(ln(re 1e-06 m-2))]
[1] 0.000000 3.542816 5.499293 6.588848 7.164921 9.083972
>
> #Standard Deviation
> sd_breaks <- classIntervals(tract_crs$density_log, n=5, style="sd")$brks
> print("Standard Deviation Breaks:")
[1] "Standard Deviation Breaks:"
> print(sd_breaks)
[1] -1.3602208  0.3782599  2.1167407  3.8552214  5.5937021  7.3321828  9.0706636 10.8091443
>
> #Jenk
> tract_crs$density_log <- as.numeric(tract_crs$density_log)
> jenks_breaks <- classIntervals(tract_crs$density_log, n=5, style="jenks")$brks
> print("Jenks Natural Breaks:")
[1] "Jenks Natural Breaks:"
> print(jenks_breaks)
[1] 0.000000 3.226198 4.540450 5.938882 6.982386 9.083972
> 
```

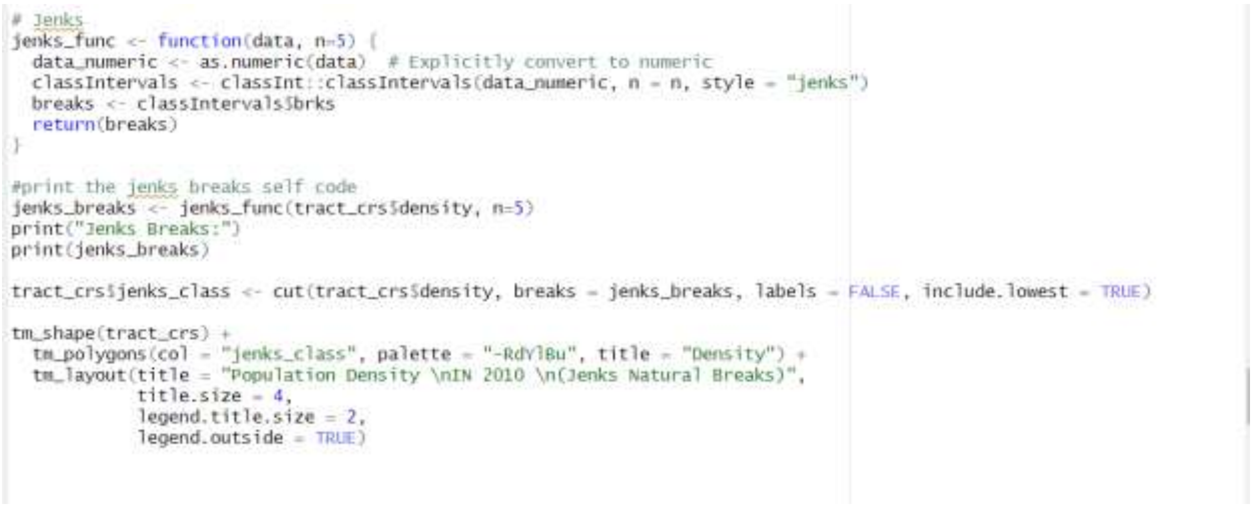
2.1 Self codings



```
    legend.outside = TRUE)

#quantile
quantile_func <- function(data, n=5) {
  quantile(data, probs = seq(0, 1, length.out = n + 1), na.rm = TRUE)
}

#print the quantile breaks self code
quantile_breaks <- quantile_func(tract_crs$density, n=5)
print("Quantile Breaks:")
print(quantile_breaks)

tract_crs$quantile_class <- cut(tract_crs$density, breaks = quantile_breaks, labels = FALSE, include.lowest = TRUE)

tm_shape(tract_crs) +
  tm_polygons(col = "quantile_class", palette = "-RdYlBu", title = "Density ") +
  tm_layout(title = "Population Density in Indiana 2010 (Quantile)",
            title.size = 4,
            legend.title.size = 2,
            legend.outside = TRUE)
```

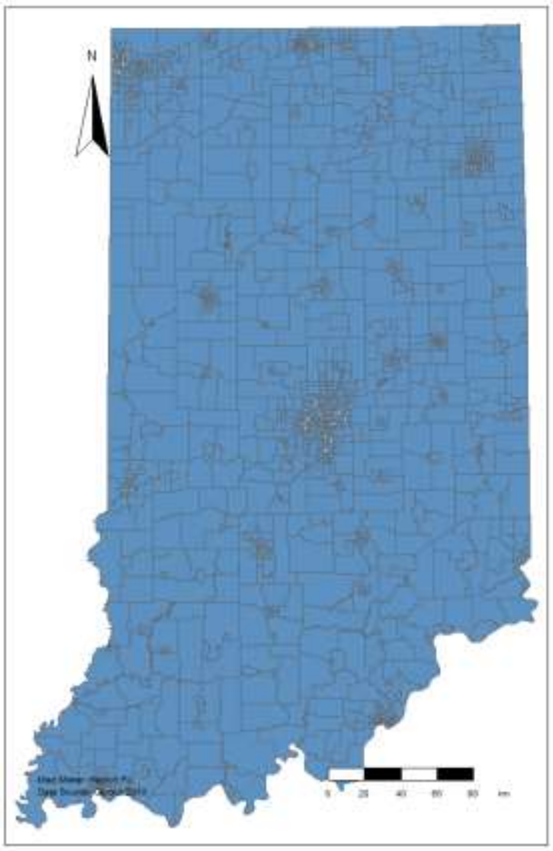


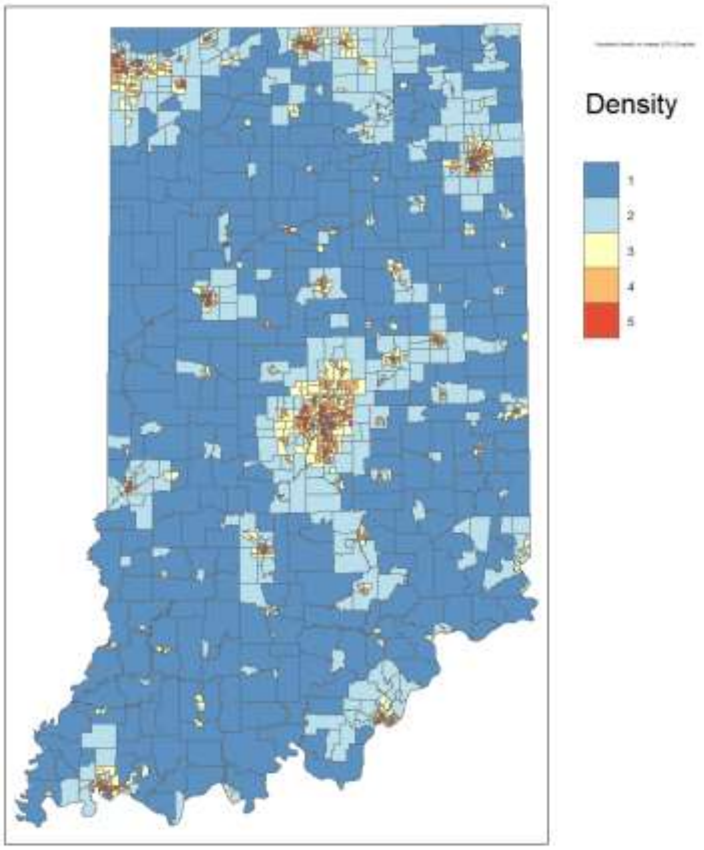
```
# Standard Deviation
standard_deviation_func <- function(data) {
  data <- as.numeric(data)  # Convert data to plain numeric
  mean_val <- mean(data, na.rm = TRUE)
  sd_val <- sd(data, na.rm = TRUE)
  seq(mean_val - 4*sd_val, mean_val + 3*sd_val, by = 1*sd_val)# tried to change the number of breaks.
}

sd_breaks <- standard_deviation_func(tract_crs$density_log)
print("Standard Deviation Breaks:")
print(sd_breaks)

tract_crs$sd_class <- cut(tract_crs$density_log, breaks = sd_breaks, labels = FALSE, include.lowest = TRUE)

# Plot the map
tm_shape(tract_crs) +
  tm_polygons(col = "sd_class", palette = "-RdYlBu", title = "Density (Log)") +
  tm_layout(title = "Population Density in Indiana 2010 (Standard Deviation)", title.size = 10, legend.outside = TRUE)


# Standard Deviation
standard_deviation_func <- function(data) {
  data <- as.numeric(data)  # Convert data to plain numeric
  mean_val <- mean(data, na.rm = TRUE)
  sd_val <- sd(data, na.rm = TRUE)
  seq(mean_val - 2*sd_val, mean_val + 10*sd_val, by = 2*sd_val)# tried to change the number of breaks.
}

sd_breaks <- standard_deviation_func(tract_crs$density)
print("Standard Deviation Breaks:")
print(sd_breaks)

tract_crs$sd_class <- cut(tract_crs$density, breaks = sd_breaks, labels = FALSE, include.lowest = TRUE)

# Plot the map
tm_shape(tract_crs) +
  tm_polygons(col = "sd_class", palette = "-RdYlBu", title = "Density") +
  tm_layout(title = "Population Density in Indiana 2010 (Standard Deviation)", title.size = 10, legend.outside = TRUE)

# Jenks
```

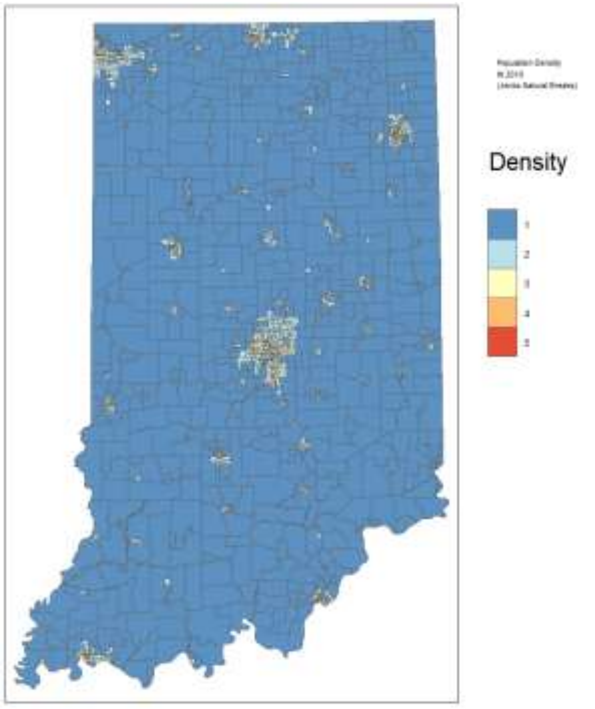


```
# Jenks
jenks_func <- function(data, n=5) {
  data_numeric <- as.numeric(data)  # Explicitly convert to numeric
  classIntervals <- classInt::classIntervals(data_numeric, n = n, style = "jenks")
  breaks <- classIntervals$brks
  return(breaks)
}

#print the jenks breaks self code
jenks_breaks <- jenks_func(tract_crs$density, n=5)
print("Jenks Breaks:")
print(jenks_breaks)

tract_crs$jenks_class <- cut(tract_crs$density, breaks = jenks_breaks, labels = FALSE, include.lowest = TRUE)

tm_shape(tract_crs) +
  tm_polygons(col = "jenks_class", palette = "-RdYlBu", title = "Density") +
  tm_layout(title = "Population Density \nIN 2010 \n(Jenks Natural Breaks)",
            title.size = 4,
            legend.title.size = 2,
            legend.outside = TRUE)
```

2.2 Self Coding Maps




Density
1
2
3
4
5


Population Density
IN 2010
(Jenks Natural Breaks)
Density
1
2
3
4
5