Thematically Mapping High Need Areas in South Carolina

STAT 540 Graduate Course Project

Rachel Passer

11/24/2015

Executive Summary

This research project explores areas of socioeconomic need in South Carolina. As my background is in geography and geographic information systems (GIS), this project will explore geographic patterns of high need in South Carolina using R. The geographic area units used are zip code tabulation areas, or ZCTAs. ZCTAs are zip code equivalent areas that work well with US Census Bureau data. For the purposes of this project, my socioeconomic indicators will be defined as the percentage of the population in poverty, the percentage that are unemployed, the percentage who lack a high school diploma, and the percentage who are uninsured. These socioeconomic indicators are measures provided by the US Census Bureau. I will be using all of the ZCTAs in the state of South Carolina as my study area. The functions created in R will input a table with the socioeconomic indicators for every ZCTA and a shapefile (a data format native to GIS software) of all the ZCTAs, merge them together to thematically map all four socioeconomic indicators with a final map showing high need areas. ZCTAs that fall within the 75th percentile or higher in all four indicators will be defined as high need areas.

Introduction

High Need in South Carolina

The reason I am researching high need areas in South Carolina is because this work strongly relates to my profession. Currently I am a researcher at the Institute for Families in Society, where I am a GIS Analyst doing Medicaid policy research for the state of South Carolina. Much of this work involves looking at socioeconomic information and how it relates to health services, in hopes of better allocating health services to those most in need. That being said, a lot of the socioeconomic information used comes from US Census Bureau's American Factfinder website. This is a website that is completely open to the public and provides datasets with various measures of the American population in tabular format. While I cannot use the health services information my office is privy to, I can explore the socioeconomic side of our research by using US Census data. Coming from a geography background, census data is highly valued and useful data as it always has a geographic component to its tables. Since this is a statistical programming course, I felt it would be interesting to apply my typical approach to socioeconomic GIS research within an R environment.

Study Area

For any project involving geographic information systems (GIS), one of the most important questions to answer early in the project is where your study area will be. The study area determines the scope of the work possible, the types of data available, and the effectiveness of the approach. For this project, I would like to identify areas of high need throughout the state of South Carolina. That means the extent of the research includes the entire state, but I will still need a smaller unit of measurement within the state to explore socioeconomic patterns. The area units of measurement I will be using are called zip code tabulation areas, or ZCTAs. They are available for download through the US Census' Topologically

Integrated Geographic Encoding and Referencing (TIGER) products. There are approximately 422 ZCTAs in South Carolina, so with that many units of measurement (and also rows in a table) it should be possible to see a variety in socioeconomic patterns throughout the state.

<u>R Libraries</u>

Several libraries were installed into R in order to prepare the data and subsequently map it. Here is a list of the libraries used and the purposes they serve:

- **rgdal**: R's interface to the popular C/C++ spatial data processing library gdal. This package allows the R user to read in and work with shapefiles.
- **tmap**: This is a newer package that rapidly creates thematic maps.
- **RColorBrewer**: This is a package of color schemes based on Cynthia Brewer's Color Brewer website. Many geographers use her color schemes for their maps. The tmap library works well with RColorBrewer.

<u>Data Analysis</u>

There are two main components to data when doing thematic mapping: geographic data, and tabular data. Geographic data means the points, lines, and polygons shown on a map, also known as the spatial component. Tabular data refers to the data that informs the geographic data how to be displayed. In order to have a thematic map, there must be geographic data teamed up with tabular data. In practice, the most common file format native to GIS software that does this is called a shapefile (.shp). A shapefile is actually composed of several files, one being the shape data, another a database file which is referred to as the attribute table, along with others such as a projection file, to ensure the shapefile is placed in the correct geographic location. To make a quick example of how this works in this project, imagine looking at all of the zip codes in South Carolina at once. There would be many little jagged polygons all over the state. Now imagine that each zip code corresponds to one row in a table. That table may contain many columns, such as the name of the zip code, its population, its area, or even perhaps a rate or a percentage. With information like that attributed to each zip code, it is possible to look at geographic patterns for each zip code by looking at the different counts, rates, or percentages by each zip code. It may even be possible to see patterns and extract further meaning from those patterns.

Often, when a shapefile is acquired, its attribute table may only list information identifying each area unit. That means the attribute table may only have a few columns, listing information like Zip Code 1, Zip Code 2, Zip Code 3, and so on. It is up to the researcher to join in pertinent data to the shapefile in order to make a thematic map with the shapefile. That is what this project will be doing in R. I have downloaded socioeconomic data from the US Census Bureau's American Factfinder website for four indicators: percent in poverty, percent unemployed, percent without a high school diploma, and the percent that are uninsured. My first function reads in the non-spatial data that gives percentages for these indicators for every ZCTA in South Carolina. These tables also include a column simply identifying the ZCTA as well. Since both the shapefile and the census table have a ZCTA identifier column, it is possible to join the census table's information to the shapefile, essentially by matching the ZCTA identifier between the census table and the shapefile. The function has commands to properly join this census data to the shapefile.

After the census data has been joined to the shapefile's attribute table, the shapefile now has a lot of different information that can be mapped from it. New information can also be obtained and calculated from the census data by ZCTA. However, now it is necessary to decide how to best display geographic information. When doing GIS work, it is necessary to look at the distribution of the data for every area unit, and think about the classification of that distribution that makes the most sense for the research. Since this project is looking to find commonality between four different indicators, classifying all indicators consistently is important. This will be accomplished by examining percentile values for each indicator. After the census data has been joined to the shapefile, the function will create objects that find 0, 25^{th}, 50^{th}, 75^{th} and 100^{th} percentile values, which will be used as the class break values when that indicator is mapped. So if an object is created to find these percentile values for the percent in poverty by ZCTA, the returned percentile values will be input as an argument into the mapping command, to inform that command of how many classes will be used, what ranges of values go in each class, and what color that class becomes on the map. If this is done the same way for every indicator, then it is possible to create four maps with a consistent methodology. At this point, everything in the first function is done: a table and a shapefile are read in, their tables are prepared for joining, they are joined, class breaks are calculated for the indicator (column) to be mapped, and the shapefile and class breaks are passed out of the function to be used in the next thematic mapping function.

<u>Thematic Mapping in R</u>

With the shapefiles prepared for mapping, the functions in the tmap library will now be put into use for the thematic mapping function. The way tmap works is by first specifying the object being used (the shapefile), and then specifying how to use shapefile, based on the column of interest. The ‘how’ commands depend on the type of data being mapped; ZCTAs are polygons, meaning each ZCTA can have a border line, with a fill in the middle. Polygons can be thematically mapped by their fill, their border lines, or both. Other ‘how’ commands can map lines, points, or pixels in a raster image. A thematic map is plotted when all the tmap functions needed have all the correct arguments filled, and all of the labels and map elements added in. Map elements such as a scale bar and a legend can be added through tmap, along with labels and titles.

Identifying High Need

The next step is identifying areas of high need based on the four indicators. A third function was written to pull information from the four indicator's results from being run through the first function. This function takes their class breaks and data, to create a new object that selects where all four indicators are greater than or equal to the 75th percentile. When all four indicators satisfy that condition, that ZCTA will be identified as a high need area. So at this point, the shapefile's attribute table includes percentages for all four indicators, along with a field called 'HighNeed' that is 'TRUE' when it meets those conditions. A subset of this shapefile can be selected out as well, where 'HighNeed' is only 'TRUE' to be mapped as well. With the entire attribute table data analyzed and prepared, everything can now be mapped.

Results

The first map (figure 1) shows the percent in poverty by ZCTA in South Carolina with a histogram. The second map (figure 2) shows the percent unemployed by ZCTA in South Carolina with a histogram. The third map (figure 3) show the Percent without a high school diploma by ZCTA in South Carolina with a histogram. The fourth map (figure 4) shows the Percent that are Uninsured by ZCTA in South Carolina with a histogram. The fifth map (figure 5) show the areas of high need by ZCTA in South Carolina. These are ZCTAs where the percentage of the socioeconomic indicator is greater than or equal to the 75th percentile, for every indicator. The accompanying table shows more detailed information for the high need ZCTAs.

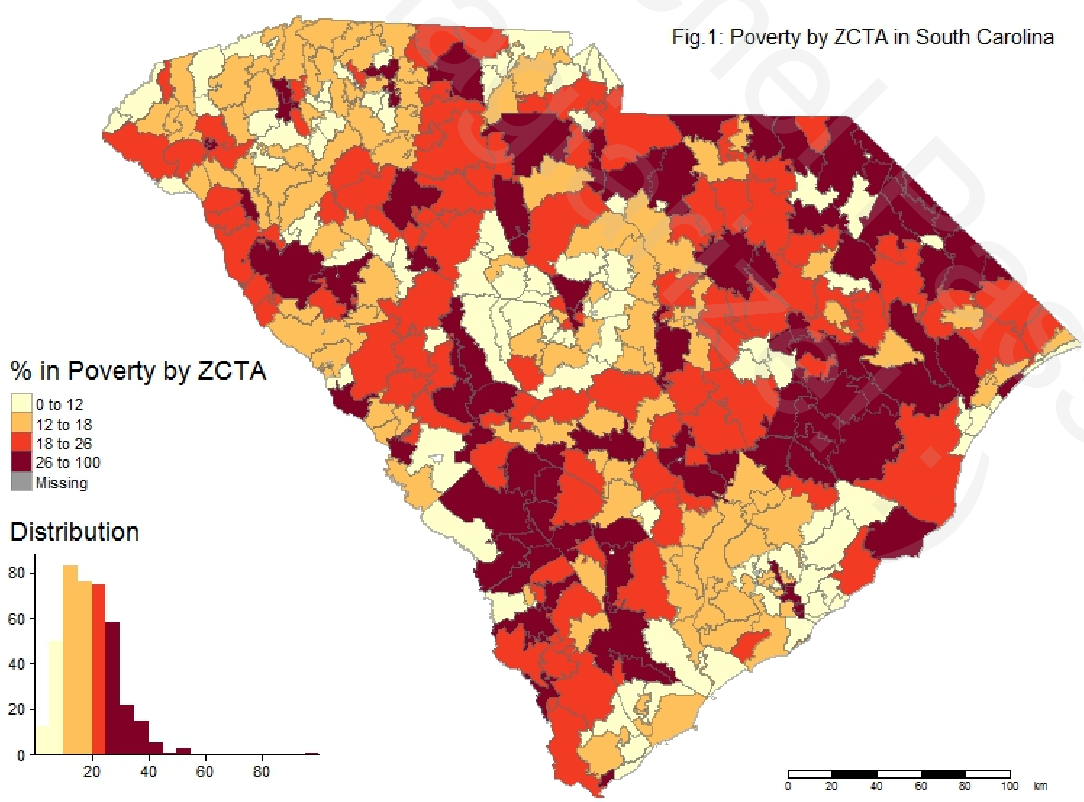


Fig.1: Poverty by ZCTA in South Carolina

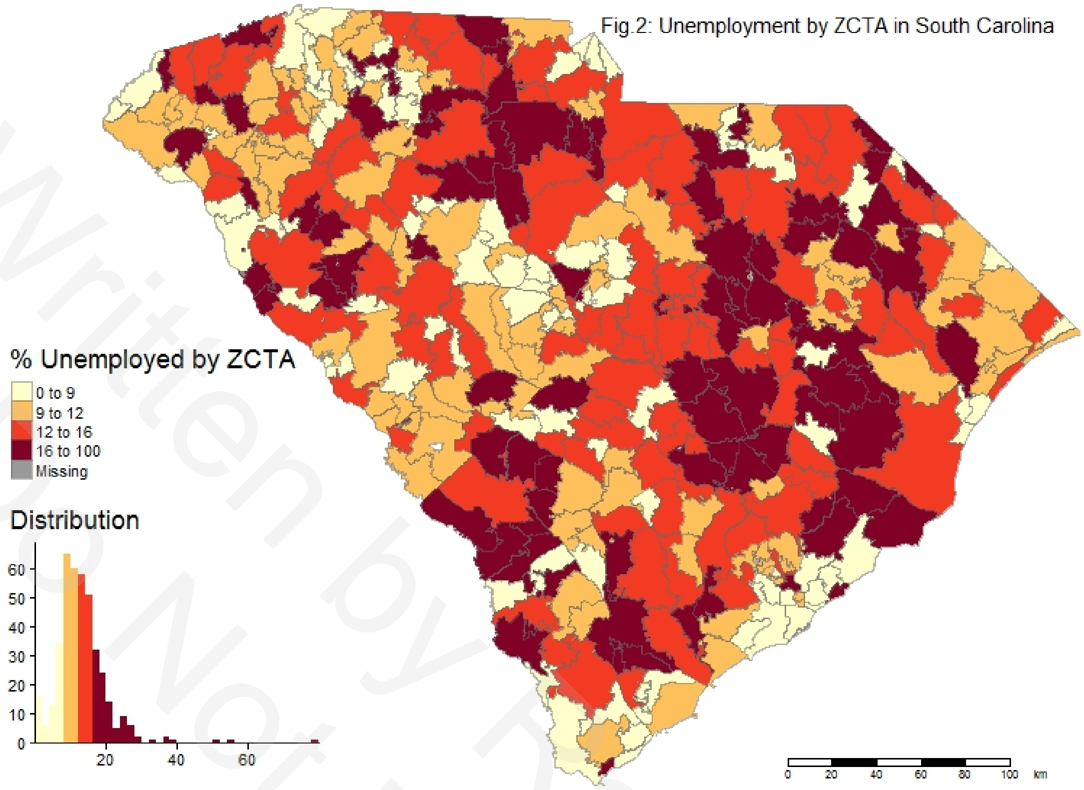


Fig.2: Unemployment by ZCTA in South Carolina

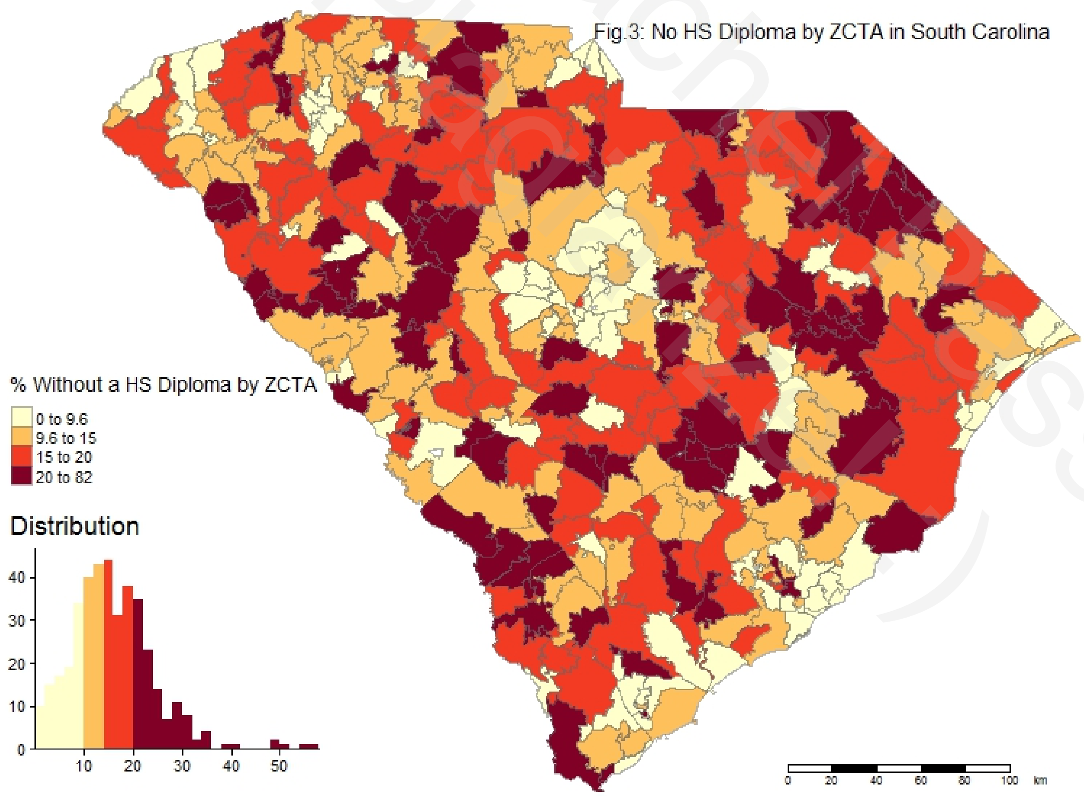


Fig.3: No HS Diploma by ZCTA in South Carolina

Fig.4: Uninsured by ZCTA in South Carolina

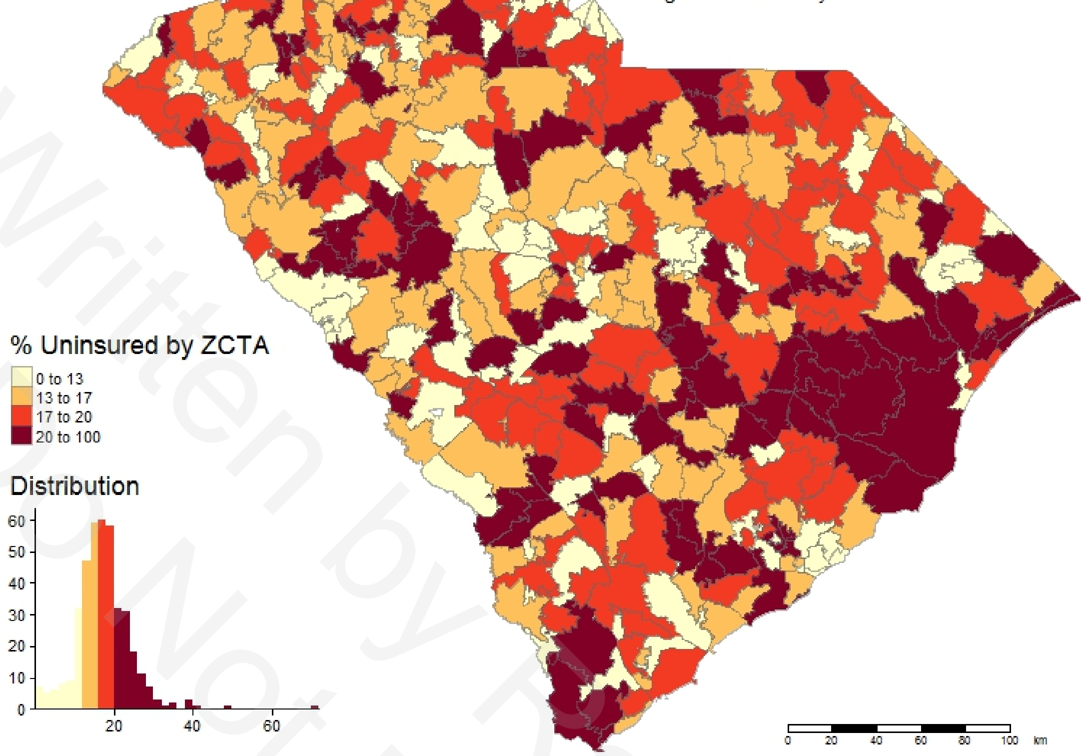
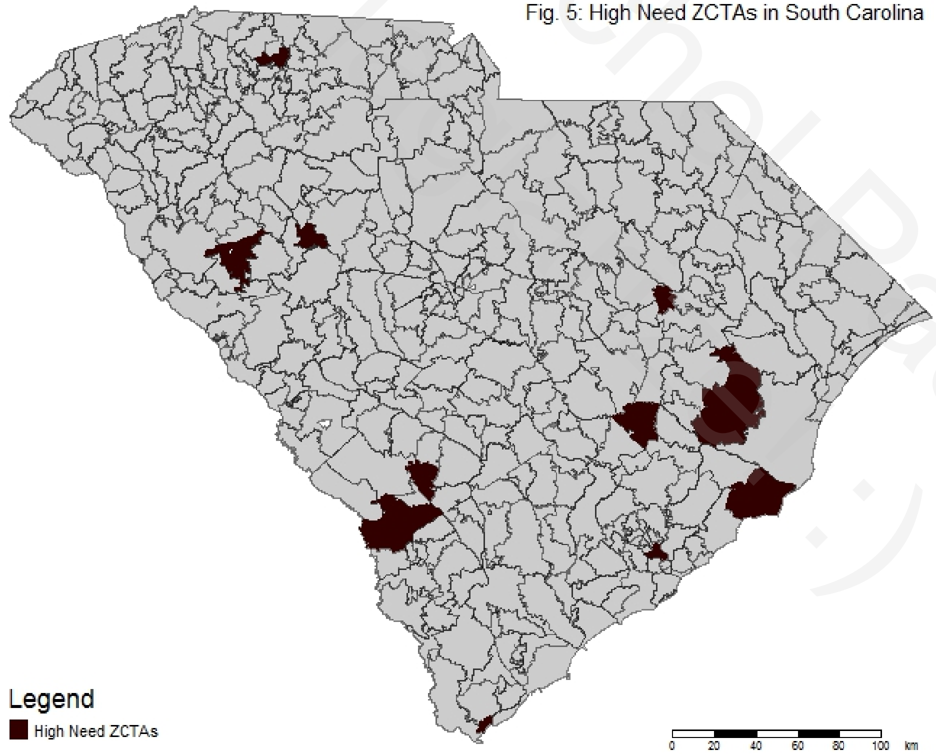


Fig. 5: High Need ZCTAs in South Carolina



ID	ZCTA	Name	Total Population	% Poverty	% Unemployed	% No HS Diploma	% Uninsured	High Need
7	29843	Olar	1,735	28.2	16.4	25.4	23.8	TRUE
11	29915	Daufuskie Island	599	51.1	23.1	50.7	21.2	TRUE
164	29580	Nesmith	1,455	27.2	19.5	21.4	24.4	TRUE
223	29405	North Charleston	24,590	35.7	17.1	23.5	22.7	TRUE
234	29114	Olanta	2,366	25.8	20.7	24.6	25.4	TRUE
275	29827	Fairfax	3,326	36.2	20.8	23.9	25.4	TRUE
280	29458	McClellanville	2,286	27.3	22.1	33.9	26.9	TRUE
314	29145	Silverstreet	664	25.9	28.4	32.8	27.8	TRUE
324	29303	Spartanburg	22,311	32.5	18.5	29.9	25.9	TRUE
333	29810	Allendale	5,093	39.7	27	21	23.2	TRUE
340	29468	Pineville	1,856	35	24.8	28.9	24	TRUE
358	29510	Andrews	10,970	34.6	16.6	22	31.6	TRUE
377	29646	Greenwood	27,982	31.4	17.3	21.2	20.6	TRUE

Discussion

<u>Patterns Observed</u>

Before interpreting the maps, I will define how I interpret meaning when I look at these maps. All of these indicators were selected together because typically, the higher the percentage of poverty, unemployment, high school dropouts, and uninsured, the data begins show places that lack services needed to live a healthy lifestyle where the population can support themselves. Areas with high percentages were interpreted as having inadequate education available or lack of support to encourage education, and culture that is deprived of the means to provide a healthy lifestyle. Areas that have low percentages for these indicators were interpreted as areas that are not having trouble educating their population; the population is able to support itself and has the means to live a healthy lifestyle. Looking at the four socioeconomic indicator maps, there is a geographic pattern that becomes apparent. It appears throughout South Carolina, there are more areas of high need in the low country and the more rural parts of the state. The areas with the lowest percentages are most common in the upstate around Greenville and Spartanburg, as well as a general concentration of lower percentages around the urban centers of the state. Based on these maps, it could be speculated that these are the areas in South Carolina that need the most help socioeconomically, in order to eventually level the playing field for every South Carolinian.

<u>Shortcomings and Further Experimentation</u>

While this project discussed how to thematically map high need in South Carolina, much more could have been done to further analyze these patterns. From a research standpoint, a different approach to mapping high need in South Carolina would be ideal. Many social scientists created indices of need, to create one measure that combines the results of many indicators in an area. Creating an index and then comparing the index to other measures such as urban and rural areas, race or ethnicity, income levels, means of transportation, patterns of disease and more could have been added to better portray what is really going on geographically in South Carolina. Census indicators were chosen because they are freely available for download, and are rather accurate and new.

As for the resulting maps, so much more could have been done within the functions to customize how they plotted. The tmap library along with several other geospatial libraries provides different mapping options, such as density maps and interpolation techniques that output a smooth surface.

As for the programming in R, different approaches to writing functions are possible; this project wrote three functions, one to prepare the data, one to map the data, and a third to calculate and map high need areas, but that can absolutely be modified. A function could be created for many different combinations of steps such as inputting a polygon shapefile, creating centroid points out of the polygons, and mapping an interpolated surface of points. Thematically mapping can be done with one large function that inputs the data and outputs a map as well. The third function was created in a way that shows how some of the steps from the first two functions can be added together as well. The more the steps are broken up, the most versatile they may be though, it all comes down to the user's preference and overall functionality.

Finally, coming from the point of view of a GIS Analyst that regularly uses ESRI's ArcGIS, while R is much more customizable with data, it takes much longer to get the intended output than it would in ArcGIS. The commands are less intuitive and require a lot more research to figure out how to string the commands together to create a thematic map containing typical map elements. R was not originally made for GIS, so it takes a lot of libraries and know-how to turn it into a GIS environment. For folks who simply want to make aesthetically pleasing maps, R may not be the best choice. On the flip side, R can also be advantageous for use in GIS research if the work is more heavily centered on data analysis, and the spatial output does not require heavy cartographic design. In my opinion, the best case scenario would be to take advantage of the extensive libraries in R, analyze the spatial data in there and export the output into ArcGIS where it could be displayed more neatly. All in all, I definitely enjoyed learning a new way to make maps and analyze spatial data. Hopefully this leads to more programming techniques being implemented into my professional GIS career.

References

Tutorials to Handle Spatial Data in R:

https://gis.stackexchange.com/questions/45327/tutorials-to-handle-spatial-data-in-r

Creating Maps in R:

https://github.com/Robinlovelace/Creating-maps-in-R

Introduction to visualizing spatial data in R:

http://robinlovelace.net/r/2014/01/30/spatial-data-with-R-tutorial.html

Package 'tmap':

https://cran.r-project.org/web/packages/tmap/tmap.pdf

Notes on Spatial Data Operations in R:

https://dl.dropboxusercontent.com/u/9577903/broomspatial.pdf

Data Sources

Download data like the South Carolina ZCTA shapefile with US Census TIGER files:

https://www.census.gov/geo/maps-data/data/tiger.html

Download US Census Bureau tables with American Factfinder (use 'Advanced Search'):

http://factfinder.census.gov/faces/nav/jsf/pages/index.xhtml

Appendix

Project Proposal

STAT 540 Graduate Student Course Project Proposal
Rachel Passer

For my graduate student project, I would like to explore areas of socioeconomic high need in South Carolina. Since I currently work as a GIS Analyst, I want to carry out a common mapping project that I would typically do in ESRI's ArcGIS, but accomplish it within R.

First, I will choose and justify my geographic units of measurement. I would like to use zip code tabulation areas (ZCTAs) from the Census Bureau because I use them often in my research, and I feel the amount of ZCTAs in South Carolina is a reasonable amount of units (roughly 400 in South Carolina) for a research project.

I will then identify and define what "high need" is in South Carolina. I will choose a small handful of socioeconomic indicators from the US Census' American Community Survey (2013 5-year estimates as they have the lowest margin of error), and compile a high-need index based on the units of measurement that fall in the highest percentile/quartile for each indicator. More specifically, if I start with the amount of South Carolinians that are uninsured according to the Census divided by the total population, I can find the 75^{th} percentile of those values, and deem values above that percentile "high need." I would then move on to the next indicator and do the same, until this approach has been carried out for each indicator. At that point, I would identify areas of high need as those that fall within the highest percentile for any of the indicators I examined. I would map each indicator, plus a combined indicator showing areas that are high need in every indicator.

As for how to carry this out in R, I would load the appropriate packages into the R environment, read in my census data csv's and my ZCTAs (GIS shapefiles, .shp file format), join them together, and do some subsets and selections to find where ZCTAs are high need in multiple indicators. Once all the fields (vectors) are populated and properly attributed to the joined data, I would make a new copy of the dataset that includes all the indicators needed for plotting. I would make subsets of that dataset that I could plot for individual indicators' high need areas. From there, I would plot the different indicators, either one at a time or all together, and manipulate the graphics settings to make them conducive to mapping.

Finally, if possible, I would like to convert my ZCTAs to ZCTA centroids (points instead of polygons), and attempt generating interpolated surfaces of high need from those centroids. This would include something like clustering, hotspots, or a simple density surface. Hopefully this would result in rasterized (smoothed, pixelated) surfaces that show areas of high need in South Carolina that I could plot.

As for how this would work in R, I envision writing two functions that can be re-used for each high need indicator. The first function would read a csv table and a shapefile (the geographic data including the lines, map projection, and table), join them together appropriately with output that creates a new joined dataset, and then create subsets based on the areas in high need. For example, this could be a subset of the data where the percentage uninsured in any ZCTA is greater than or equal to the 75th percentile of the distribution. This function would simply read, assemble and query the data, and the output could be some summary data about the joined geographic data and its subsets, along with a simple plot showing demonstrating the PDF of the distribution. The plot could point out where the 75th percentile is, so the reader understands that the following maps the actual values (in this case, percentage) that the 75th percentile would represent.

I would then make a mapping function. This function would take the output geographic datasets from the first function and appropriately plot them. It could plot side-by-side a choropleth map by quartiles (so the 75th percentile ZCTAs would be the highest quartile and have their own color), and then perhaps next to that plot, an interpolated surface showing the same thing.

I would then do a report to explain this whole process, along with my references for learning GIS & R capabilities.

R Functions

```
> shp_prep(csv, shapefile_dsn, shp_name, indicator, join_col, merge_old_name,
merge_name)

function(csv, shapefile_dsn, shp_name, indicator, join_col, merge_old_name,
merge_name)
{

    ################# This is a function that will read in a csv, get quartile class
break values from the indicator column,
    ################# The function will return a prepared shapefile for mapping, as
well as the class breaks needed to classify the indicator by quartiles.
    ################# This function will first prep the csv data so that it can be
joined to geographic data (a shapefile, specifically)
    ################# by the ZCTA column. When the csv data with the indicator data is
joined to the shapefile, it can then be mapped.

    #load the rgdal library, this library reads spatial data like shapefiles
    library(rgdal)

    # read in csv
    # no geographic component to this data yet.

    SES_table <- read.csv(csv, header = TRUE)

    # make sure indicator field is numeric.

    SES_table[, indicator] <- as.numeric(levels(SES_table[, indicator]))[SES_table[,
indicator]]

    # Before joining this csv to a shapefile for mapping,
    # we need to know the data distribution, so we know the class breaks to input to
the mapping functions.

    # Get class breaks by creating vectors from quantile results.
    # run quantile function on indicator field to get class breaks for the maps.

    breaks <- quantile(SES_table[, indicator], c(0, .25, .5, .75, 1), na.rm = FALSE,
names = TRUE, type = 7)

    # Read in a shapefile, the geographic component
    # This requires the dsn (the folder filepath the shapefile is in), and the name of
the shapefile (do not type extensions like .shp, just the name)
    # Only put in the shapefile name and not the extension because most shapefiles can
be composed of 3-8 files.

    shapefile <- readOGR(shapefile_dsn, shp_name)

    # GIS data, namely a shapefile, cannot be used and viewed with the same common R
commands.
    # In order to access the data (or as GIS folks say, the "attribute table")
associated with the shapefile, the data slot must
    # be specified like "shapefile@data". If you just use the command "shapefile", it
will be read as something to plot.

    # What fields between the table and the shapefile are you planning to join on?
    # For this join approach, they must have a column in common. There are many ways
to join data, but this one was most
```

```
    # effective and only uses base R commands. The columns being joined on must have
the same column name in both the shapefile and table.

    # also make sure the data type between these two fields is the same. one may be a
character and one may be factor.
    # Coerce table's join field to factor if it's not already, so it matches the
shapefile's join field name, and data type

    SES_table[, join_col] <- as.factor(SES_table[, join_col])

    # Rename the table's column name so it matches the shapefile's column name

    names(SES_table) <- sub(merge_old_name, merge_name, names(SES_table))

    # Now we can join
    #shapefile@data <- left_join(shapefile@data, SES_table) is another potential way
to join, did not work for me (from the dplyr package).

    shapefile@data <- data.frame(as(shapefile, "data.frame"),
SES_table[match(shapefile@data[, merge_name],SES_table[, merge_name]),])

    # make a new shapefile that is appropriately named to reflect the SES data merged
in

    Output_shp <- shapefile


    # pass the shapefile and class breaks out of the function

    map_data <- list(Output_shp, breaks)
    #names(map_data) = c("Output Shapefile", "Class Breaks")

    map_data

}
```

```
>thematic_map(function1_info, indicator, color_scheme, classes,
classification, legend_title, map_title)
```

```
function (function1_info, indicator, color_scheme, classes, classification,
legend_title, map_title){

    # This function will take a shapefile (.shp, a file format composed of up to six
files) from the shp_prep function
    # and pass it into the tmap commands, resulting in a map plot.

    # The map plot will classify the data in quartiles, but the quartiles are based on
the
    # results from the other DataPrep function.

    # load the R libraries needed

    library(rgdal) #reads spatial data
    library(tmap) # thematic mapping functions
    library(RColorBrewer) # Cynthia Brewer's color schemes for use in R, based on her
website colorbrewer2.org


    # get data from the shp_prep function into this function

    data_prep <- function1_info

    # the shapefile was first in the list of output from the data prep function, pull
it and make a new object

    shapefile <- data_prep[[1]]

    # get class breaks, which was the second output object output from the data prep
function

    class_breaks <- data_prep[[2]]

    # now we have the data prep object we need, the thematic mapping can begin!

    # note on thematic mapping in R:
    # If you want to quickly generate a map, the command 'qtm' (quick thematic map)
will
    # plot a very basic map with minimal arguments. If you want to have more
    # control of the data classifications, line widths, color schemes and more, the
best way
    # is to use a combination of tmap commands strung together. For more information
on this
    # I highly recommend reading through Martijn Tennekes' documentation on the tmap
package, here:
    # https://cran.r-project.org/web/packages/tmap/tmap.pdf

    Map <- tm_shape(shapefile) +
        tm_polygons(indicator, palette = color_scheme, n = classes, style =
classification, breaks = class_breaks, title = legend_title, cex = 0.9, legend.hist =
TRUE, legend.hist.title = "Distribution", border.alpha = .5) +
        tm_layout(title = map_title, title.size = .96, inner.margins = c(0, 0.1, 0,
0), outer.margins = 0.01, draw.frame = FALSE, title.position = c("right", "top")) +
        tm_scale_bar()

    # This results in a thematic map.

    # tm_shape specifies the shapefile to thematically map
    # tm_polygons does the thematic map for polygons (which is what the ZCTAs are)
    # tm_scale_bar makes a scale bar
```

```
    # tm_layout adds other general map elements to the plot
    # string them all together and you get a map layout

    # plot the map as output

    Map

}
```

```
>high_need(ind1, ind2, ind3, ind4)

function(ind1, ind2, ind3, ind4) {

    ################# This is a function that passes in the output data from the four
indicators, in order to
    ################# calculate where all indicators are >= the 75th percentile,
indicating areas of high need
    ################# in several different socieconomic indicators. This may show
areas of high need.
    ################# However, this function should be generic enough that it could be
easily modified to take results from several
    ################# different types of data at varying values and output a value in
a new field. However, a shapefile is a requirement
    ################# to map anything.

    # this function requires the tmap library, which provides many thematic mapping
functions
    library(tmap)


    # Get the output from the first function for each indicator, that means the
results from the four runs of the first function
    # Assign them appropriately

    Indicator1 <- ind1
    Indicator2 <- ind2
    Indicator3 <- ind3
    Indicator4 <- ind4

    # This function only needs one shapefile, but it needs the class breaks from all
four indicators
    # Extract the breaks output from each of the four indicator runs and make them
objects in this function

    breaks1 <- ind1[[2]]
    breaks2 <- ind2[[2]]
    breaks3 <- ind3[[2]]
    breaks4 <- ind4[[2]]

    # Extract the 75th percentile value as a new object for each of the for indicators

    ind1_75th <- breaks1[[4]]
    ind2_75th <- breaks2[[4]]
    ind3_75th <- breaks3[[4]]
    ind4_75th <- breaks4[[4]]

    # Pick one shapefile and make it an object in this function

    High_Need_shp <- Indicator3[[1]]

    # Check shapefile indicator fields and make sure they are all numeric and not
factor
    # Coerce to numeric

    High_Need_shp@data[, 5] <- as.numeric(levels(High_Need_shp@data[,
5]))[High_Need_shp@data[, 5]]
    High_Need_shp@data[, 6] <- as.numeric(levels(High_Need_shp@data[,
6]))[High_Need_shp@data[, 6]]

    # Now take all rows in the table that are >= to the highest quartile in every
indicator.
    # so if highest quartile in all 4, it's high need
```

```
    High_Need_shp@data$HighNeed <- ((High_Need_shp@data[, 5] >= ind1_75th) &
(High_Need_shp@data[, 6] >= ind2_75th) & (High_Need_shp@data[, 7] >= ind3_75th) &
(High_Need_shp@data[, 8] >= ind4_75th))

    High_Need_Only_shp <- High_Need_shp[High_Need_shp@data$HighNeed == TRUE,]

    # The following section of code will use the tmap library to do thematic mapping.

    # input a shapefile, output a map

    # There are many ways to make a thematic map in R. The amount of commands gets
longer
    # the more you choose to customize what you see on the map.
    # If you want to quickly generate a map, the command 'qtm' (quick thematic map)
will
    # throw up a very basic map with minimal customizations. If you want to have more
    # control of the data classifications, line widths, color schemes and more, the
best way
    # is to use a combination of tmap commands strung together. For more information
on this
    # I highly recommend reading through Martijn Tennekes' documentation on tmap,
here:
    # https://cran.r-project.org/web/packages/tmap/tmap.pdf

    # This function will map the high need ZCTAs throughout South Carolina with a
solid fill.

    Map <- tm_shape(High_Need_shp) +
        tm_polygons(col = "grey80", labels = c("High Need ZCTA", "Not High Need"), lwd
= 1.5, border.col = "black", border.alpha = .5) +
        tm_layout(title="Fig. 5: High Need ZCTAs in South Carolina", title.size = .95,
draw.frame = FALSE, title.position = c("right", "top")) +
        tm_scale_bar() +
        tm_shape(High_Need_Only_shp) +
        tm_fill("HighNeed", palette = "#330000", title = "Legend", labels = "High Need
ZCTAs")

    # This results in a map with two layers; one of the whole state by ZCTA, but not
themtically mapped, just specified to show up as grey.
    # The second layer is a shapefile that only contains the ZCTAs where HighNeed =
TRUE, shown in burgundy.

    # tm_shape specifies the shapefile to thematically map
    # tm_polygons does the thematic map for polygons (which is what the ZCTAs are)
    # tm_polygons is actually a combination of tm_borders and tm_fill
    # tm_fill specifies what information is displayed in the fill of the polygons
    # tm_scale_bar makes a scale bar
    # tm_layout adds other general map elements to the plot
    # string them all together and you get a map layout

    # Access the attribute table of the High Need shapefile for viewing purposes

    High_Need_table <- High_Need_Only_shp@data

    # Some summary statistics for the High Need ZCTAs

    High_Need_sum <- summary(High_Need_Only_shp@data)

    # The information about to be sent out of the function

    High_Need_Out <- list(Map, High_Need_table, High_Need_sum)
```

```
    # This function outputs a Map of High Need ZCTAs, the associated attribute table,
and some summary statistics of the attribute table.

    High_Need_Out


}
```