Coding Camp: Final Coding Project

Area of Opportunity Index Analysis

Noah Fischer

9/17/2021

BACKGROUND

The New Hampshire Housing Finance Authority (New Hampshire Housing) is a quasi-public agency responsible for administering many federal and state affordable housing programs each year. One of the most important programs that they are responsible for is the allocation of the state's yearly low-income housing tax credits (LIHTCs) to qualified developers for the construction or preservation of affordable rental housing.

Part of the authority's responsibility is to ensure that construction of affordable housing does not become overly concentrated in particular areas. Over concentration of affordable housing can perpetuate instances of poverty and segregate communities based on wealth, and inextricably, race. In order to avoid this problem, some housing finance authorities have taken to developing an evaluative tool called an Area of Opportunity Index. The index that I develop in this analysis is at the census tract level, and takes into account variables such as median income, poverty rate, and educational attainment.

METHODOLOGY

First, we will load the necessary libraries. These include 'tidyverse', 'tidycensus', and 'tmap'. 'Tidyverse' will allow us to run necessary statistical analysis and clean the data. 'Tidycensus' will allow us to import ACS data via an API directly from the Census Bureau database. 'Tmap' will allow us to create static and active maps to display results.

```r
library(tidyverse)
library(tidycensus)
library(tmap)
```

Next, we need to decide on the variables to use for the analysis. The Census codes variables, so we can create an object that contains all of the variables available to us from the 2019 5-year ACS dataset.

```r
acs_vars <- load_variables(2019, "acs5")
```

We then create an object that contains a vector of the variables that we want to take from the Census. You can name each variable so that when we import the data they do not appear as a series of numbers and letters.

```r
variables <- c(total_poverty = "B06012_002",
               total_pop = "B01003_001",
               med_inc = "B06011_001",
               bach_degree = "B06009_005",
               grad_degree = "B06009_006"
)
```

The ‘get_acs’ function (part of the ‘tidycensus’ package) can then be used to collect this data via an API.

```
census_data <- get_acs(geography = "tract",
                       state = "NH",
                       variables = variables,
                       survey = "acs5",
                       year = 2019,
                       output = "wide",
                       geometry = T) %>%
  select(GEOID, NAME, total_povertyE, total_popE, med_incE,
         bach_degreeE, grad_degreeE, geometry)
```

```
##   |                                                                              |
```

The output = “wide” argument will import the data with one observation per column. The geometry = T argument will create an additional column that lists the geometric points of each census tract so that we can map out the state later on.

First, we will calculate the percentage of people in each census tract that live below 100% of the poverty line.

```
census_data <- census_data %>% mutate(pct_poverty =
                                        (total_povertyE/total_popE)*100)
```

For the purposes of this analysis, we will consider high educational attainment as having attained a Bachelor’s degree or more. We can calculate the percentage of residents in each tract that meet this educational attainment standard.

```
census_data <- census_data %>% mutate(high_education =
                              ((bach_degreeE + grad_degreeE)/total_popE) *100)
```

ANALYSIS

First, we will standardize each observation by creating z-scores for each variable.

```
census_data <- census_data %>%
  mutate(pov_z = (pct_poverty - mean(pct_poverty, na.rm=T))/ sd(pct_poverty, na.rm=T))

census_data <- census_data %>%
  mutate(educ_z = (high_education - mean(high_education, na.rm=T))/ sd(high_education, na.rm=T))

census_data <- census_data %>%
  mutate(inc_z = (med_incE - mean(med_incE, na.rm=T))/ sd(med_incE, na.rm=T))
```

We now have to set cutoff points to determine if the z-score falls within a range that we determine as being a good indicator of an area of opportunity or not. We can first use the ‘summary’ function to see the different quartiles of the new z-score variables.

```
summary(census_data$pov_z)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
## -1.2422 -0.6926 -0.2739  0.0000  0.4430  5.5621       3
```

```
summary(census_data$inc_z)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
## -3.3625 -0.6314 -0.0632  0.0000  0.6591  3.5266       3 
```

```
summary(census_data$educ_z)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's 
## -1.99965 -0.73698 -0.02813  0.00000  0.70465  2.95166        3 
```

Because we want to minimize the amount of poverty, we will say that the census tract will receive a 1 if its poverty z-score is in the 25th percentile or lower (z -0.6926). Contrastingly, we want higher median incomes and higher rates of educational attainment, so we will say that census tracts with z-scores that are in the 75th percentile or higher (z 0.6591 for income, z 0.70465 for education) will receive a one. The total index score is simply the sum of the three individual variable index scores.

```
census_data <- census_data %>%
              mutate(poverty_index_score =
                    ifelse(pov_z <= -0.6926, 1, 0),
          educ_index_score = ifelse(educ_z >= 0.70465, 1, 0),
          inc_index_score = ifelse(inc_z >= 0.6591, 1, 0),
          total_index_score = poverty_index_score + educ_index_score + inc_index_score)
```

For some reason, the data contains a row that does not contain a gemoetry and therfore cannot be graphed. We will remove that row to continue with the analysis.

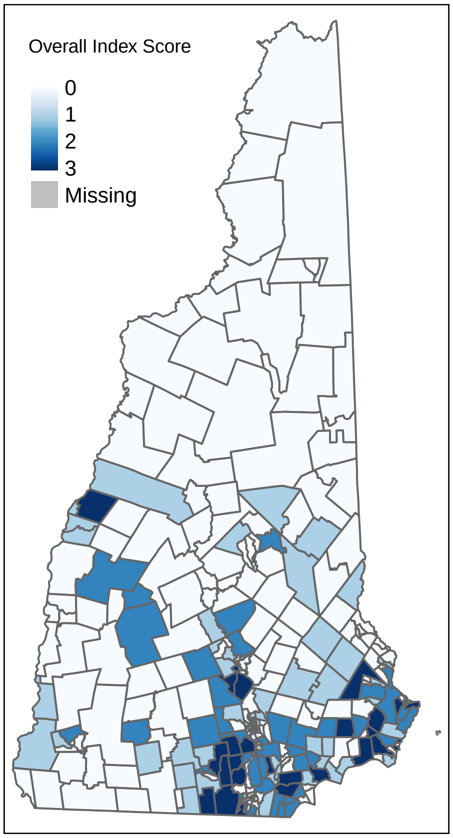
```
census_data <- census_data[-295,]
```

RESULTS

Now that we have created the overall index scores based on the demographic variables, we can map the state's census tracts and see which areas can be deemed high opportunity areas.

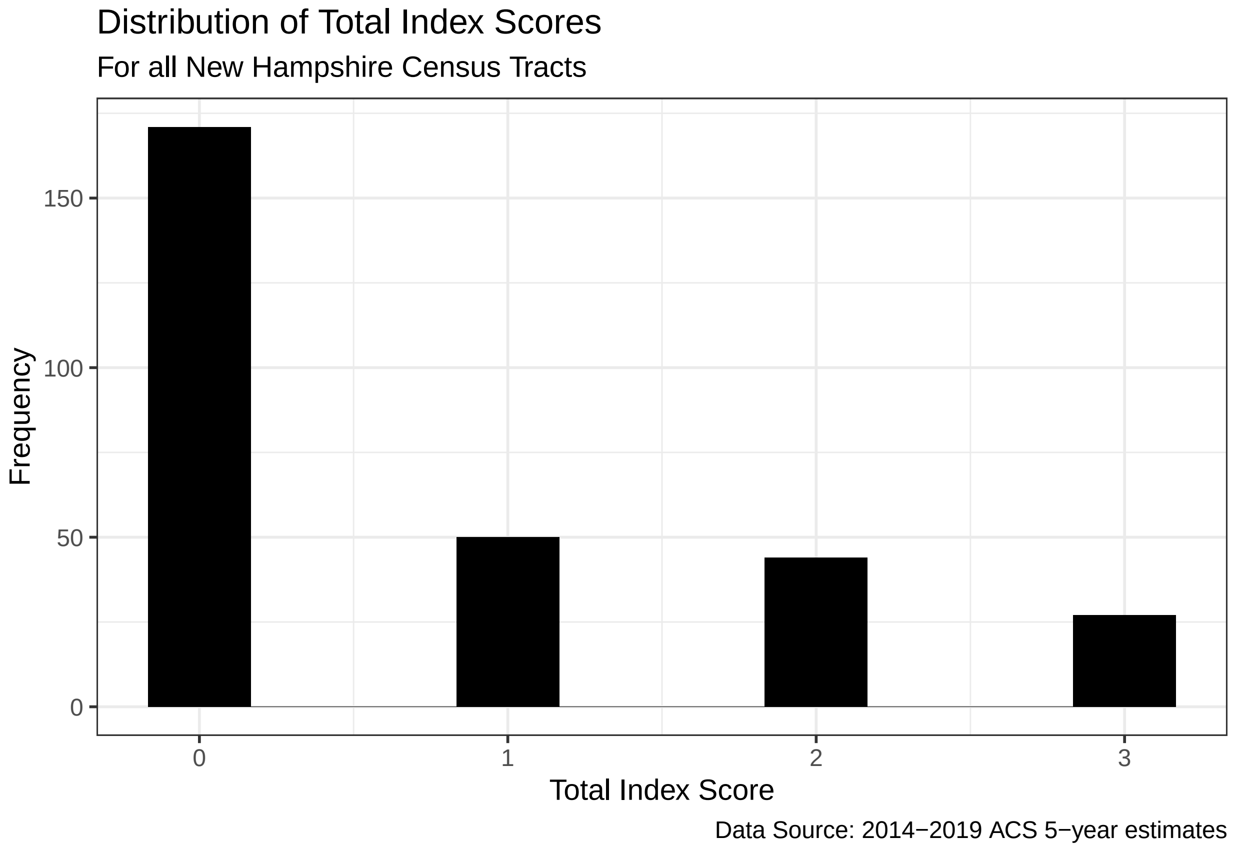
```
tract_opp_map <- tm_shape(census_data) +
    tm_polygons(col="total_index_score", title="Overall Index Score", palette="Blues",
                            breaks = c(0,1,2,3),
                style="cont")

tract_opp_map
```



We can also use ggplot to create a historgram of the total index scores.

```
ggplot(data=census_data, mapping = aes(x=total_index_score)) +
  geom_histogram(fill = "black", bins = 10) +
  labs(x="Total Index Score", y="Frequency", title="Distribution of Total Index Scores", subtitle = "Fo
  theme_bw()
```



Finally, we can create a summary table of the census tracts that had an overall index score of 3. Because the geometry variable is "sticky" it is hard to remove it from the data. I could not figure out how to remove it so that it would not show up in the summary table. I tried pulling it out and also tried subsetting the original dataframe to not include the variable. None of these attempts were successful.

```r
summary_table <-
  census_data %>%
  filter(total_index_score == 3) %>%
  summarise(average_poverty = mean(pct_poverty, na.rm=T),
            average_pct_high_education = mean(high_education, na.rm=T),
            average_median_income = mean(med_incE, na.rm=T))

summary_table %>%
  knitr::kable()
```

average_poverty	average_pct_high_education	average_median_income	geometry
2.009898	40.40176	51917.96	MULTIPOLYGON (((-70.81811 4...