Extracting Demographic Indexes for the Casco Bay Watershed

July 27, 2022

Contents

Introduction

CBEP, like other National Estuary Programs will receive additional funding to support our programs via the "Bipartisan Infrastructure Law" signed into law last December.

EPA has recently released guidance for applying for those funds. A core component of the guidance is that overall, the NEP program should comply with the White House's "Justice 40" initiative, which requires that "at least 40% of the benefits and investments from BIL funding flow to disadvantaged communities."

EPA suggested that we use the National-scale EJSCREEN tools to help identify "disadvantaged communities" in our region. The EPA guidance goes on to suggest we focus on five demographic indicators:

- Percent low-income;
- Percent linguistically isolated;
- Percent less than high school education;
- Percent unemployed; and
- Low life expectancy.

This notebook builds on the work in "Calc_Indexes.pdf" to calculate data for the Casco Bay Watershed Census block groups, and calculates how Casco Bay Census block groups compare at National, Statewide, and Regional scales.

Load Libraries

```
library(tidyverse)
#> -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
#> v ggplot2 3.3.6     v purrr   0.3.4
#> v tibble  3.1.7     v dplyr   1.0.9
#> v tidyr   1.2.0     v stringr 1.4.0
#> v readr   2.1.2     v forcats 0.5.1
#> -- Conflicts ------------------------------------------ tidyverse_conflicts() --
#> x dplyr::filter() masks stats::filter()
#> x dplyr::lag()    masks stats::lag()
library(readr)
```

Set Graphics Theme

This sets `ggplot()`graphics for no background, no grid lines, etc. in a clean graphic format.

```
theme_set(theme_classic())
```

Load Data

Folder References

I use folder references to allow limited indirection, thus making code from GitHub repositories more likely to run "out of the box".

```
data_folder <- "Original_Data"
#dir.create(file.path(getwd(), 'figures'), showWarnings = FALSE)
```

I use the "Original_Data" folder to retain data in the form originally downloaded. That minimizes the chances of inadvertently modifying the source data. All data was accessed via EJScreen. The 2021 EJSCREEN Data was accessed on July 26, 2022, at https://gaftp.epa.gov/EJSCREEN/2021/. I downloaded geodatabases, and open the geospatial data they contained in ArcGIS and exported the tabular attribute data to CSV files. That tabular CSV data is provided in the "Original Data" folder here.

Load Data

The Tabular data is quite extensive, which poses some data access challenges. The raw CSV file contains more than 200,000 records, and more than 100 columns. Most, but not all are numeric.

```
the_file <- "casco_block_groups.csv"
the_path <- file.path(data_folder, the_file)

cb_data <- read_csv(the_path, n_max = 81,
                    col_types = c(paste(c(rep('-',2),  'c-',  rep('d',2),
                                          rep('-',7), 'd'), collapse = '')))
```

```
the_file <- "National_Draft_Indexes.csv"
the_data <- read_csv(the_file,
                     col_types = paste0('ccc', rep('d', 21)))
```

```
cb_data<- cb_data %>%
  select(ID) %>%
  inner_join(the_data, by = 'ID')
```

Save Data

```
write_csv(cb_data, "cb_block_groups_indexes.csv")
```

Calculate Thresholds

We have six different indexes, and we want threshold values for each at National, State, and Casco Bay Region levels. It's convenient to automate the calculations using a small function and the `map()` function.

```
(nms <- names(cb_data)[19:24])
#> [1] "Index_Raw"         "Index_5_Ptiles"    "Index_4_Ptiles"
#> [4] "Index_best_Ptiles" "PCA_Index_V1"      "PCA_Index_V2"
```

Utility Function

This function calculates the 80th percentile (by default, anyway) of a named data column from a data frame. There is no error checking, so this is NOT appropriate for programming with out more work.

```
quantile_select_col <- function(.data, .col, .q = 0.8) {
  return(quantile(.data[,.col], .q, na.rm = TRUE))
}

the_data %>%
  quantile_select_col( "Index_Raw")
#>      80%
#> 31.15984
```

Calculations

I calculate a named vector of threshold values at National, State and Casco Bay Regional levels.

```
National <- map(nms, function(x) quantile_select_col(the_data, x))
National <- unlist(National) # Flatten List to numeric vector
names(National) <- nms       # Add Names
National
#>         Index_Raw    Index_5_Ptiles    Index_4_Ptiles Index_best_Ptiles
#>          31.15984          69.86830          71.20022          71.34104
#>      PCA_Index_V1      PCA_Index_V2
#>          73.68687         154.05754
```

```
Maine <- map(nms,
             function(x) quantile_select_col(the_data[the_data$STATE_NAME == 'Maine',], x))
Maine <- unlist(Maine) # Flatten List to numeric vector
names(Maine) <- nms       # Add Names
Maine
#>         Index_Raw    Index_5_Ptiles    Index_4_Ptiles Index_best_Ptiles
#>          26.68095          59.35638          59.25347          59.41836
#>      PCA_Index_V1      PCA_Index_V2
#>          62.97380         131.38726
```

```
Region <- map(nms, function(x) quantile_select_col(cb_data, x))
Region <- unlist(Region) # Flatten List to numeric vector
names(Region) <- nms       # Add Names
Region
#>         Index_Raw    Index_5_Ptiles    Index_4_Ptiles Index_best_Ptiles
#>          23.79057          50.40002          56.61339          58.68217
#>      PCA_Index_V1      PCA_Index_V2
#>          55.08407         111.22754
```

```
thresholds <- bind_rows(National, Maine, Region, .id = 'Scale') %>%
  mutate(Scale = c('National', 'Maine', 'Region'))
```

Calculate NUmber of Threshold Exceedences

I use a functional programming approach for calculating whether specific Casco Bay Census block groups exceed each threshold. Here I pass both a dataframe and a names list or vector containing the thresholds.

Utility Function

```
threshold_compare <- function(.data, .thresholds, .col) {
  return(.data[.col] > .thresholds[.col])
}
```

```
National_Exceeds <- map(as.list(nms),
                        function(x) threshold_compare(cb_data, National, x))
National_Exceeds <- as.data.frame(National_Exceeds)
names(National_Exceeds) <- nms
unlist(map(National_Exceeds, sum, na.rm = TRUE))
#>         Index_Raw    Index_5_Ptiles    Index_4_Ptiles Index_best_Ptiles
#>                 2                 0                 5                 3
#>      PCA_Index_V1      PCA_Index_V2
#>                 2                 0
```

```
Maine_Exceeds <- map(as.list(nms),
                     function(x) threshold_compare(cb_data, Maine, x))
Maine_Exceeds <- as.data.frame(Maine_Exceeds)
names(Maine_Exceeds) <-  nms
unlist(map(Maine_Exceeds, sum, na.rm = TRUE))
#>         Index_Raw    Index_5_Ptiles    Index_4_Ptiles Index_best_Ptiles
#>                 4                 8                12                14
#>     PCA_Index_V1     PCA_Index_V2
#>                3                4
```