SDOH Measures for ZCTA, ACS 2017-2021

RobertLupo

2024-05-28

## This project aims to analyze Social Determinants of Health (SDOH) measures for ZIP Code Tabulation Areas (ZCTA) using the American Community Survey (ACS) data from 2017 to 2021. By cleaning and exploring the dataset, we seek to uncover trends and insights that can inform data-driven decision-making. The analysis employs various statistical techniques and predictive modeling to evaluate the impact of different measures on health outcomes.

### Introduction

## Social Determinants of Health (SDOH) are non-medical factors that influence health outcomes. These include socioeconomic status, education, neighborhood and physical environment, employment, social support networks, and access to healthcare. This project focuses on analyzing SDOH measures across various ZIP Code Tabulation Areas (ZCTAs) using data from the American Community Survey (ACS) between 2017 and 2021. The goal is to identify key factors that impact health outcomes and understand the geographical distribution of these determinants.

### Data Description

## The dataset used in this project is sourced from the American Community Survey (ACS) and provides SDOH measures for ZIP Code Tabulation Areas (ZCTAs) from 2017 to 2021. The dataset includes variables such as data\_value, margin of error (moe), total population, latitude, longitude, and various categorical indicators. The data was cleaned to handle missing values and ensure consistency across measures.

### Methodology

## The methodology for this project includes several steps:  
## 1. Data Cleaning: Handling missing values and ensuring data consistency.  
## 2. Exploratory Data Analysis (EDA): Generating summary statistics and visualizations to understand data distributions and relationships.  
## 3. Correlation Analysis: Identifying relationships between numerical variables.  
## 4. Predictive Modeling: Developing models to predict health outcomes based on SDOH measures. Models considered include linear regression, logistic regression, and support vector machines.  
## 5. Model Evaluation: Assessing model performance using metrics such as R-squared, mean squared error, and accuracy, and employing cross-validation techniques.

### Results

## The analysis revealed several key insights:  
## - The distribution of key numerical variables such as data\_value, moe, and total population.  
## - Correlation analysis showed significant relationships between data\_value, latitude, longitude, and total population.  
## - Predictive modeling identified important SDOH measures that influence health outcomes. The models were evaluated and validated to ensure reliability.

### Discussion/Interpretation of Results

## The findings suggest that certain SDOH measures have a significant impact on health outcomes. For instance, areas with higher total populations and specific geographical locations tend to have different health outcomes. These insights can inform public health strategies and resource allocation to address disparities. However, the analysis is limited by the quality and granularity of the data, and further research could expand on these findings by incorporating additional variables and more granular data.

### Conclusion

## In conclusion, this project highlights the importance of SDOH measures in understanding health outcomes. By analyzing ACS data from 2017 to 2021, we identified key factors that influence health and provided insights that can guide public health decision-making. Future research could explore more detailed datasets and consider additional variables to further refine these findings.

### References

## American Community Survey (ACS) data, 2017-2021.

#EDA  
# Load necessary libraries  
required\_packages <- c("data.table", "naniar", "ggcorrplot", "maps", "ggplot2", "dplyr")  
  
# Install missing packages  
installed\_packages <- rownames(installed.packages())  
for (pkg in required\_packages) {  
 if (!(pkg %in% installed\_packages)) {  
 install.packages(pkg)  
 }  
}  
  
# Load libraries  
lapply(required\_packages, library, character.only = TRUE)

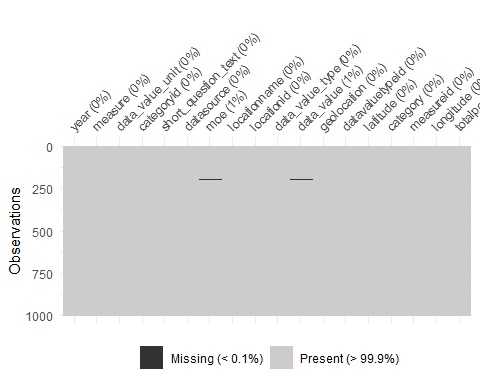
## [[1]]  
## [1] "maps" "ggcorrplot" "naniar" "data.table" "lubridate"   
## [6] "forcats" "stringr" "dplyr" "purrr" "readr"   
## [11] "tidyr" "tibble" "ggplot2" "tidyverse" "stats"   
## [16] "graphics" "grDevices" "utils" "datasets" "methods"   
## [21] "base"   
##   
## [[2]]  
## [1] "maps" "ggcorrplot" "naniar" "data.table" "lubridate"   
## [6] "forcats" "stringr" "dplyr" "purrr" "readr"   
## [11] "tidyr" "tibble" "ggplot2" "tidyverse" "stats"   
## [16] "graphics" "grDevices" "utils" "datasets" "methods"   
## [21] "base"   
##   
## [[3]]  
## [1] "maps" "ggcorrplot" "naniar" "data.table" "lubridate"   
## [6] "forcats" "stringr" "dplyr" "purrr" "readr"   
## [11] "tidyr" "tibble" "ggplot2" "tidyverse" "stats"   
## [16] "graphics" "grDevices" "utils" "datasets" "methods"   
## [21] "base"   
##   
## [[4]]  
## [1] "maps" "ggcorrplot" "naniar" "data.table" "lubridate"   
## [6] "forcats" "stringr" "dplyr" "purrr" "readr"   
## [11] "tidyr" "tibble" "ggplot2" "tidyverse" "stats"   
## [16] "graphics" "grDevices" "utils" "datasets" "methods"   
## [21] "base"   
##   
## [[5]]  
## [1] "maps" "ggcorrplot" "naniar" "data.table" "lubridate"   
## [6] "forcats" "stringr" "dplyr" "purrr" "readr"   
## [11] "tidyr" "tibble" "ggplot2" "tidyverse" "stats"   
## [16] "graphics" "grDevices" "utils" "datasets" "methods"   
## [21] "base"   
##   
## [[6]]  
## [1] "maps" "ggcorrplot" "naniar" "data.table" "lubridate"   
## [6] "forcats" "stringr" "dplyr" "purrr" "readr"   
## [11] "tidyr" "tibble" "ggplot2" "tidyverse" "stats"   
## [16] "graphics" "grDevices" "utils" "datasets" "methods"   
## [21] "base"

# Increase memory limit (Windows only)  
if (.Platform$OS.type == "windows") {  
 memory.limit(size = 16000)  
}

## Warning: 'memory.limit()' is no longer supported

## [1] Inf

# Load the dataset  
data <- fread("bumh-rgsq\_version\_37.csv")  
  
# Check for missing data in a sample of the dataset  
vis\_miss(dplyr::slice\_sample(data, n = 1000))



# Fill missing values with the mean of their respective 'measure' groups  
data[, c("data\_value", "moe") := lapply(.SD, function(x) {  
 mean\_val <- mean(x, na.rm = TRUE)  
 ifelse(is.na(x), mean\_val, x)  
}), .SDcols = c("data\_value", "moe")]  
  
# Verify that there are no missing values left  
missing\_counts <- data[, lapply(.SD, function(x) sum(is.na(x)))]  
  
# Print missing\_counts  
print(missing\_counts)

## year measure data\_value\_unit categoryid short\_question\_text datasource  
## <int> <int> <int> <int> <int> <int>  
## 1: 0 0 0 0 0 0  
## moe locationname locationid data\_value\_type data\_value geolocation  
## <int> <int> <int> <int> <int> <int>  
## 1: 0 0 0 0 0 0  
## datavaluetypeid latitude category measureid longitude totalpopulation  
## <int> <int> <int> <int> <int> <int>  
## 1: 0 0 0 0 0 0

# Exploratory Data Analysis (EDA)  
# Display basic information about the dataset  
glimpse(data)

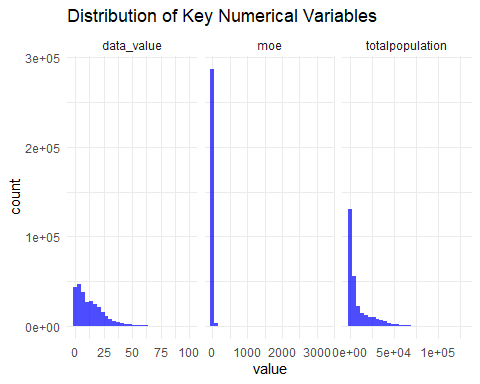
## Rows: 291,024  
## Columns: 18  
## $ year <chr> "2017-2021", "2017-2021", "2017-2021", "2017-2021"…  
## $ measure <chr> "Persons of racial or ethnic minority status", "No…  
## $ data\_value\_unit <chr> "%", "%", "%", "%", "%", "%", "%", "%", "%", "%", …  
## $ categoryid <chr> "SDOH", "SDOH", "SDOH", "SDOH", "SDOH", "SDOH", "S…  
## $ short\_question\_text <chr> "Racial or ethnic minority status", "No high schoo…  
## $ datasource <chr> "5-year ACS", "5-year ACS", "5-year ACS", "5-year …  
## $ moe <dbl> 9.400000, 4.100000, 5.600000, 9.205388, 5.000000, …  
## $ locationname <int> 97020, 95665, 97362, 97373, 95338, 97466, 98843, 9…  
## $ locationid <int> 97020, 95665, 97362, 97373, 95338, 97466, 98843, 9…  
## $ data\_value\_type <chr> "Percentage", "Percentage", "Percentage", "Percent…  
## $ data\_value <dbl> 32.400, 5.300, 20.700, 14.528, 24.400, 3.400, 2.30…  
## $ geolocation <chr> "POINT (-122.8388419 45.2226509)", "POINT (-120.65…  
## $ datavaluetypeid <chr> "Percent", "Percent", "Percent", "Percent", "Perce…  
## $ latitude <dbl> 45.22265, 38.39518, 45.06201, 45.05800, 37.50701, …  
## $ category <chr> "SDOH", "SDOH", "SDOH", "SDOH", "SDOH", "SDOH", "S…  
## $ measureid <chr> "REMNRTY", "NOHSDP", "NOHSDP", "CROWD", "POV150", …  
## $ longitude <dbl> -122.8388, -120.6528, -122.7565, -122.7764, -119.9…  
## $ totalpopulation <int> 1200, 4650, 4174, 191, 10726, 1301, 2066, 10169, 1…

# Summary statistics for numerical columns  
summary(data[, .SD, .SDcols = sapply(data, is.numeric)])

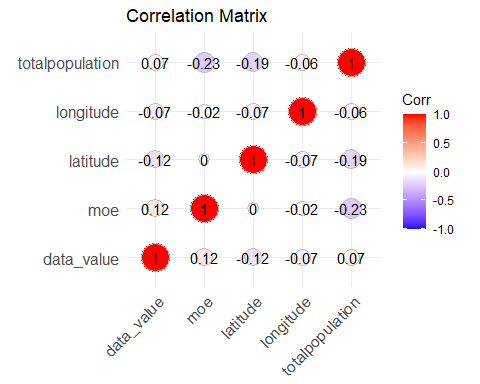
## moe locationname locationid data\_value   
## Min. : 0.000 Min. : 1001 Min. : 1001 Min. : 0.00   
## 1st Qu.: 2.600 1st Qu.:27249 1st Qu.:27249 1st Qu.: 3.90   
## Median : 4.800 Median :49761 Median :49761 Median : 10.70   
## Mean : 9.205 Mean :49742 Mean :49742 Mean : 14.53   
## 3rd Qu.: 9.800 3rd Qu.:72012 3rd Qu.:72012 3rd Qu.: 20.50   
## Max. :3290.900 Max. :99929 Max. :99929 Max. :100.00   
## latitude longitude totalpopulation   
## Min. :19.07 Min. :-176.67 Min. : 50   
## 1st Qu.:35.46 1st Qu.: -97.14 1st Qu.: 778   
## Median :39.54 Median : -88.20 Median : 2955   
## Mean :38.90 Mean : -91.02 Mean : 10196   
## 3rd Qu.:42.12 3rd Qu.: -80.32 3rd Qu.: 14012   
## Max. :71.25 Max. : -67.01 Max. :130352

# Visualize the distribution of key numerical variables  
data[, .(data\_value, moe, totalpopulation)] %>%  
 melt(measure.vars = c("data\_value", "moe", "totalpopulation"), variable.name = "variable", value.name = "value") %>%  
 ggplot(aes(x = value)) +  
 geom\_histogram(bins = 30, fill = "blue", alpha = 0.7) +  
 facet\_wrap(~variable, scales = "free\_x") +  
 theme\_minimal() +  
 labs(title = "Distribution of Key Numerical Variables")

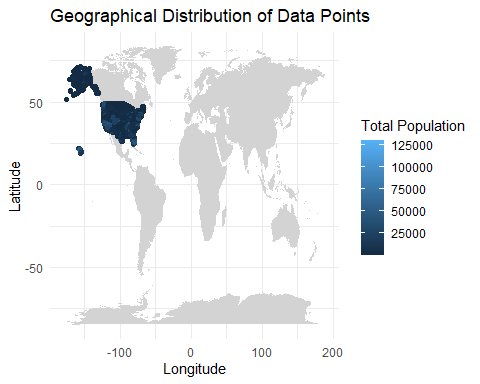
## Warning in melt.data.table(., measure.vars = c("data\_value", "moe",  
## "totalpopulation"), : 'measure.vars' [data\_value, moe, totalpopulation] are not  
## all of the same type. By order of hierarchy, the molten data value column will  
## be of type 'double'. All measure variables not of type 'double' will be coerced  
## too. Check DETAILS in ?melt.data.table for more on coercion.



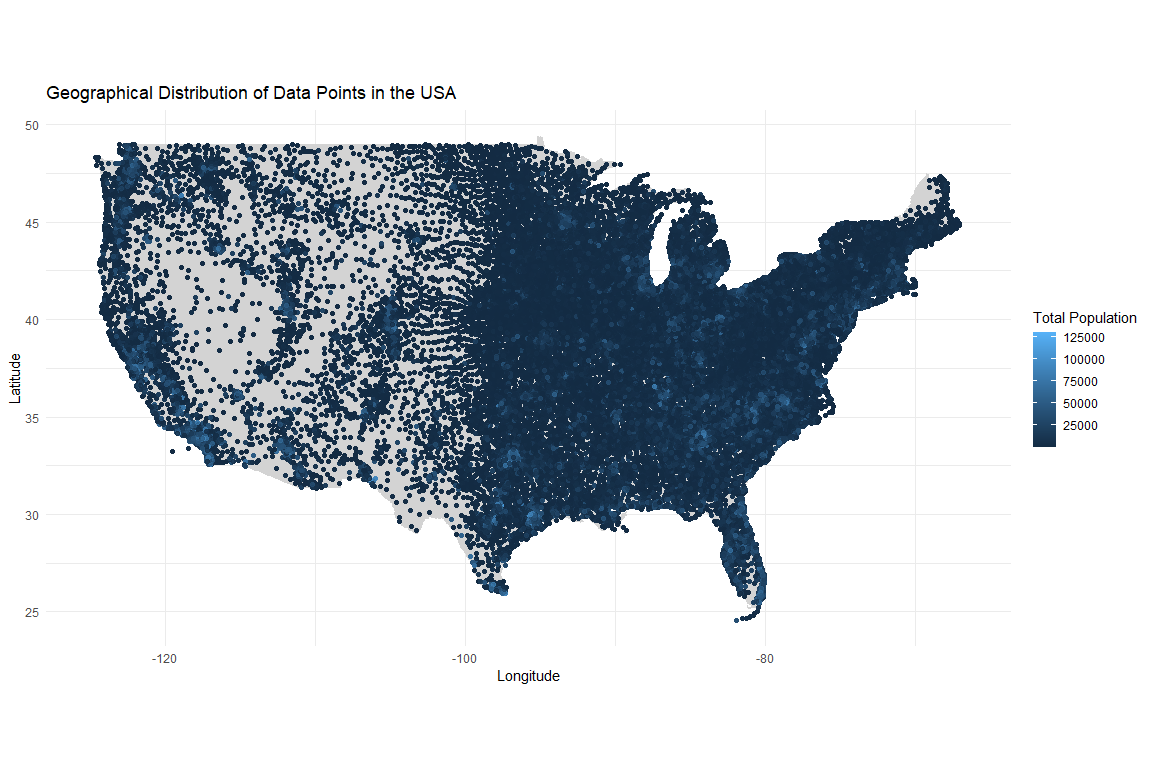
# Correlation matrix  
correlation\_matrix <- data[, .(data\_value, moe, latitude, longitude, totalpopulation)] %>%  
 cor(use = "complete.obs")  
  
# Visualize the correlation matrix  
ggcorrplot(correlation\_matrix, method = "circle", lab = TRUE) +  
 labs(title = "Correlation Matrix")



# Visualize geographical distribution  
world\_map <- map\_data("world")  
ggplot() +  
 geom\_polygon(data = world\_map, aes(x = long, y = lat, group = group), fill = "lightgray") +  
 geom\_point(data = data, aes(x = longitude, y = latitude, color = totalpopulation), alpha = 0.7) +  
 theme\_minimal() +  
 labs(title = "Geographical Distribution of Data Points",  
 x = "Longitude", y = "Latitude", color = "Total Population")



# Visualize geographical distribution focused on the USA  
usa\_map <- map\_data("state")  
ggplot() +  
 geom\_polygon(data = usa\_map, aes(x = long, y = lat, group = group), fill = "lightgray") +  
 geom\_point(data = data, aes(x = longitude, y = latitude, color = totalpopulation), alpha = 0.7) +  
 coord\_fixed(1.3, xlim = c(-125, -66.5), ylim = c(24.5, 49.5)) +  
 theme\_minimal() +  
 labs(title = "Geographical Distribution of Data Points in the USA",  
 x = "Longitude", y = "Latitude", color = "Total Population")



# Violin plot for 'data\_value' across different 'measure'  
data[, .(measure, data\_value)] %>%  
 ggplot(aes(x = measure, y = data\_value, fill = measure)) +  
 geom\_violin() +  
 theme\_minimal() +  
 labs(title = "Data Value Across Different Measures",  
 x = "Measure", y = "Data Value") +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1))

