

Comparative Examples of Geocoding and Mapping Techniques in SAS® and R Using Centers for Disease Control Survey Data

Joshua J. Cook, M.S. DS, M.S. CRM, ACRP-PM, CCRC, University of South Carolina (USC)

Louise S. Hadden, B.A., Abt Global Inc.

Swann A. Adams, Ph.D, M.S., FACE, University of South Carolina (USC)

ABSTRACT

The Centers for Disease Control (CDC) maintains the Behavioral Risk Factor Surveillance System (BRFSS), which is a rich source of survey data for public health and epidemiological research. BRFSS collects data in all 50 states as well as the District of Columbia and three U.S. territories, and includes data on health-related risk behaviors, chronic health conditions, and use of preventive services. Geospatial analysis is a crucial tool in public health and epidemiology, offering insights into geographic patterns and trends. This presentation aims to provide a comprehensive comparison of geocoding and mapping techniques using SAS® and R, utilizing data from the CDC BRFSS. By demonstrating how to geocode and visually map public health data, we seek to empower researchers and practitioners with the skills necessary to conduct spatial analyses. In this session, we will illustrate the process of geocoding and mapping BRFSS data using SAS® PROC GEOCODE and PROC SGMAP procedures, alongside equivalent methods in R, using packages such as tidygeocoder, ggplot2, and plotly. Attendees will gain insights into the strengths and limitations of each approach, understanding how to choose the appropriate tools for their specific needs. Key topics will include an overview of the CDC BRFSS dataset and its relevance to public health, step-by-step guidance on geocoding and generating spatial coordinates, techniques for creating informative and aesthetically pleasing maps, with direct comparisons between SAS® and R in terms of functionality, ease of use, and output quality. By the end of this lecture, participants will be equipped with practical knowledge to apply geocoding and mapping techniques in their work, enhancing their ability to analyze spatial data and make data-driven decisions in public health.

INTRODUCTION

THE BEHAVIORAL RISK FACTOR SURVEILLANCE SYSTEM (BRFSS)

- CDC survey/data design ([CDC - BRFSS](#))
- Variable groups
- Variables of interest

PUBLIC HEALTH AND EPIDEMIOLOGIC APPLICATIONS

- Example use cases
- Paper subset of data
- SAS and R in public health and epidemiology

GEOCODING & MAPPING

Geocoding is the process of converting addresses or other geographic identifiers into spatial coordinates, typically latitude and longitude. This transformation allows health data tied to specific locations—like residences, businesses, pharmacies, health providers, or community centers—to be placed on a map. The accuracy of geocoding can significantly impact the quality of subsequent spatial analyses, as it determines how precisely these locations are represented geographically. This tool can illuminate such socioeconomic challenges such as food and/or pharmacy deserts.

Mapping involves the visualization of geocoded data, often integrating multiple data layers to produce thematic maps. These maps can display variations in health-related factors, such as disease incidence, access to healthcare facilities, environmental exposures, or social determinants of health across different

geographic areas. By enabling visual assessment of these factors, mapping helps to identify spatial trends, clusters, or anomalies.

methods using sas®

SAS® PROCEDURES AND OTHER OPTIONS

SAS® has an amazing arsenal of tools to use and display geographic information that is relatively unknown and underutilized. We intend to highlight both new and existing capacities for creating stunning, informative maps as well as using geographic data in other ways. The two primary sources of mapping capacity in SAS are SG procedures (SGMAP and SGPLOT, as well as ATTRMAPs and annotate tools; and SAS/GRAPH, which evolved from SAS GIS many years ago. Almost all of the tools described below were originally provided to users through SAS/GRAPH; however, since SAS/GRAPH requires a separate license, many procedures and techniques were moved to BASE SAS so all users can access the geographic tools. As noted above, we will discuss custom mapping of geographic areas with SAS, as well as R, and compare and contrast.

There is an underutilized resource on the SAS website to help users make choices of how to use SAS for custom mapping and geocoding, SAS Maps Online. We strongly urge interested users to check this resource out, if nothing else to obtain geocode data and free code samples. [\[Insert Screen Shots\]](#).

SAS introduced a version of ODS GRAPHICS, a graphics system which was template based, following soon after the Output Delivery System was released. In ODS GRAPHICS, many statistical procedures could produce one or more graphics, depending on options within the procedure, etc., simply by turning on the ODS GRAPHICS option. SAS 9.2 introduced the SG family of procedures. These procedures assist users in the creation of effective graphics displaying information from statistical procedures, by using statements which invoke statistical procedures. As with ODS, SAS developed templates for each procedure and option which work seamlessly with the ODS style templates and destinations, and other SAS standards, creating attractive and meaningful graphics quickly and efficiently “out of the box”, with little effort by the programmer.

The initial SG procedures included PROC SGPLOT (single-celled graph, with many options), PROC SGSCATTER (paneled graphs with scatter plots), and PROC SGPANEL (classification panels). All these procedures support by processing and paneling when applicable. The first MAPPING use of SG procedures was PROC SGPLOT with a Polygon statement (analogous to GGLOT2), and then PROC SGMAP came into the picture.

As you might guess, there are both similarities and differences between the two systems. SG procedures are completely layout based, while SAS/GRAPH is device driven, which makes sense, since graphics were literally sent to devices. My very first SAS paper created SAS/GRAPH maps that were sent to a very expensive color printer. Because the SG procedures are available to all users, SAS examples in the R & SAS comparison will be conducted with the SG procedures.

No set up is required to use the SURVEY, BASE SAS mapping and SG procedures in SAS. However, as with R, it is necessary to set up formats, color schemas, etc. in order to produce attractive maps. While R uses geocoding to establish a link between geographic areas on a map and in the data, SAS provides some special tools for state and county level mapping in the US. SAS provides geocoding capacity for finer areas down to congressional districts and street address coding, as well as other geographic entities.

A discussion of SAS procedures relating to geographic presentations of data follow below.

PROC GEOCODE:

- This procedure is used for geocoding—translating addresses into geographical coordinates (latitude and longitude). It supports various methods of geocoding, including street-level, ZIP code, city, and area-level geocoding. PROC GEOCODE is available with Base SAS.

[Code snippet showing syntax]

PROC SGMAP:

- Introduced in SAS 9.4M3, this procedure is part of the ODS Graphics procedures and is used for creating modern, high-quality maps. It supports various layers including scatter and series plots on maps, and it can integrate with PROC GEOCODE output for mapping geocoded data. There is a noticeable reduction in the lines of code necessary to generate a map.
 - %sganno - SAS-written annotation macros for SG procedures
 - attrmaps - allows the assignment of color, shapes, and more to data values

Like PROC GEOCODE and all SG procedures, PROC SGMAP is available with Base SAS.

[Code snippet showing syntax]

PROC MAPIMPORT:

- This procedure is used to import external geographic data formats into SAS datasets. It can read various formats like shapefiles (SHP), which are commonly used to store geolocation data. PROC MAPIMPORT is included with BASE SAS.

[Code snippet showing syntax]

PROC GPROJECT:

- Used for projecting geographic data from one coordinate system to another. It's useful in adjusting data to a common projection before mapping, ensuring that spatial relationships are maintained. It also allows users to project data in corollary data sets to match the projection of the map data set(s) being used. This procedure is available in BASE SAS.

[Code snippet showing syntax]

PROC GREMOVE:

- This procedure is used to remove internal borders from map areas, which can simplify the appearance of a map and make it easier to understand, especially when dealing with complex geographic boundaries. This procedure is available in BASE SAS.

[Code snippet showing syntax]

PROC GMAP:

- This procedure, a precursor to PROC SGMAP, is used to produce thematic maps such as choropleth maps, where areas are shaded in proportion to the measurement of the statistical variable being displayed on the map, such as population density or disease incidence. This procedure is a part of SAS/Graph, a separate SAS module, and is NOT available in BASE SAS.

PROC GSLIDE:

- Though not directly related to mapping, this procedure can be used to create slide presentations that include maps and other graphical outputs from SAS. This is part of SAS/GRAPH, a separate SAS module, and is NOT available in BASE SAS.

GEOCODING BY EXAMPLE WITH BRFS

TBD - Louise will add using the same variables and weighting as Josh.

MAPPING BY EXAMPLE WITH BRFS

TBD - Louise will add using the same variables and weighting as Josh, but different base maps provided by SAS.

OPEN-SOURCE METHODS USING R

R PACKAGES

FOR GEOCODING (EQUIVALENT TO PROC GEOCODE):

tidygeocoder: Tidygeocoder makes getting data from geocoding services easy. A unified high-level interface is provided for a selection of supported geocoding services and results are returned in tibble (dataframe) format.

ggmap: This package allows for geocoding and reverse geocoding through various services like Google Maps, OpenStreetMap, etc. It can translate addresses into coordinates and vice versa.

tmapprools: Provides geocoding capabilities as part of broader spatial data processing tools.

FOR THEMATIC AND OTHER TYPES OF MAPPING (EQUIVALENT TO PROC GMAP AND PROC SGMAP):

ggplot2: While primarily a general visualization package, it can be used in conjunction with `geom_polygon()` and other geoms for creating thematic maps.

tmap: Dedicated to creating thematic maps, supporting static and interactive visualizations.

leaflet: Best for creating interactive web maps. It can handle various layers and customization, making it very flexible.

sf: Provides support for simple features, a standardized way to encode spatial vector data. It can be used with ggplot2 and leaflet for mapping.

FOR IMPORTING GEOGRAPHIC DATA (EQUIVALENT TO PROC MAPIMPORT):

rgdal: Offers facilities to read and write a wide range of vector and raster geographic data formats (including shapefiles) using the GDAL (Geospatial Data Abstraction Library) and OGR data access libraries.

sf: Also capable of reading and writing various spatial data formats including shapefiles.

FOR PROJECTING GEOGRAPHIC DATA (EQUIVALENT TO PROC GPROJECT):

sp: Supports spatial data types and includes utilities for transformations and projections of geographic data.

rgdal: Also handles data projection and transformation.

FOR REMOVING INTERNAL BORDERS (NO DIRECT EQUIVALENT IN R, BUT SIMILAR FUNCTIONALITY CAN BE ACHIEVED):

Custom functions can be written using `sf` or `sp` to dissolve boundaries and merge geographic areas based on certain criteria.

FOR CREATING PRESENTATIONS (EQUIVALENT TO PROC GSLIDE):

Quarto: Useful for creating dynamic reports and presentations, where maps and other R outputs can be integrated.

officer: Allows the creation of PowerPoint presentations and can include various outputs including maps.

HANDLING BRFSS SURVEY DATA

```
library(here)
library(tidyverse)
library(survey)

# Reading in the 2022 BRFSS data
data_2022 <- readRDS(here("temp_data", "2022_data.rds"))
str(data_2022)
```

```

colnames(data_2022)

# Remove underscores from all variable names as they may cause issues in
subsequent processing
data_2022 <- data_2022 %>%
  rename_with(~str_remove(., "_"), starts_with("_"))

# Create a lookup table for state FIPS codes based on the 2022 codebook
state_fips <- tibble(
  "STATE" = c(1, 2, 4, 5, 6, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39,
40, 41, 42, 44, 45, 46, 47, 48, 49, 50, 51, 53, 54, 55, 56, 66, 72, 78),
  state = as.factor(c("Alabama", "Alaska", "Arizona", "Arkansas",
"California", "Colorado", "Connecticut", "Delaware", "District of
Columbia", "Florida", "Georgia", "Hawaii", "Idaho", "Illinois", "Indiana",
"Iowa", "Kansas", "Kentucky", "Louisiana", "Maine", "Maryland",
"Massachusetts", "Michigan", "Minnesota", "Mississippi", "Missouri",
"Montana", "Nebraska", "Nevada", "New Hampshire", "New Jersey", "New
Mexico", "New York", "North Carolina", "North Dakota", "Ohio", "Oklahoma",
"Oregon", "Pennsylvania", "Rhode Island", "South Carolina", "South Dakota",
"Tennessee", "Texas", "Utah", "Vermont", "Virginia", "Washington", "West
Virginia", "Wisconsin", "Wyoming", "Guam", "Puerto Rico", "Virgin
Islands")))
)

# Join BRFSS data with the state FIPS lookup table to add state names
data_2022 <- data_2022 %>%
  left_join(state_fips, by = "STATE")

# Verify the levels of the state factor to ensure correct matching
levels(data_2022$state)

# Create a survey design object accounting for the complex survey design
data_2022_survey <- svydesign(
  id = ~PSU, # Primary Sampling Units (PSUs)
  strata = ~STSTR, # Stratification variable
  weights = ~WT2RAKE, # Weights variable
  data = data_2022, # Data set
  nest = TRUE # Indicates nesting of the design
)

# Recode the CNCRDIF variable as a factor with descriptive labels for each
level
data_2022$CNCRDIF <- factor(data_2022$CNCRDIF, levels = c(1, 2, 3, 7, 9),
  labels = c("One", "Two", "Three or more",
"Don't know/Not Sure", "Refused"))

# Set survey options to adjust for lonely PSUs, which may have a single
observation
options(survey.lonely.psu = "adjust")

# Recreate the survey design object with adjusted survey options
data_2022_survey_adjusted <- svydesign(
  id = ~PSU,
  strata = ~STSTR,
  weights = ~WT2RAKE,
  data = data_2022,

```

```

    nest = TRUE
  )

# Calculate weighted COUNTS for the cancer type variable (CNCRDIF) by
state
cancer_counts_by_state <- svyby(
  ~CNCRDIF,                # Outcome variable
  ~STATE,                  # Grouping variable (State)
  data_2022_survey_adjusted, # Survey design object
  svytotal,                # Function to compute weighted totals
  na.rm = TRUE              # Remove missing values
)

# Reshape data for easier merging
cancer_counts_long <- cancer_counts_by_state %>%
  pivot_longer(cols = starts_with("CNCRDIF"), names_to = "Cancer_Type",
values_to = "Count") %>%
  mutate(Type = case_when(
    str_detect(Cancer_Type, "One") ~ "One",
    str_detect(Cancer_Type, "Two") ~ "Two",
    str_detect(Cancer_Type, "Three.or.more") ~ "Three or more",
    TRUE ~ NA_character_
  )) %>%
  filter(!is.na(Type)) %>%
  select(-Cancer_Type)

# Join BRFSS data with the state FIPS lookup table to add state names
cancer_counts_long <- cancer_counts_long %>%
  left_join(state_fips, by = "STATE")

# Calculate the "At least one" count per state by summing counts for "One,"
"Two," and "Three or more"
at_least_one_counts <- cancer_counts_long %>%
  group_by(STATE) %>%
  summarise(AtLeastOneCount = sum(Count, na.rm = TRUE))

# Calculate the total number of "At least one" cases in the U.S.
total_us_cases <- sum(at_least_one_counts$AtLeastOneCount, na.rm = TRUE)

# Calculate the proportion of "At least one" cases for each state relative
to the U.S. total
at_least_one_counts <- at_least_one_counts %>%
  mutate(USPercent = (AtLeastOneCount / total_us_cases) * 100)

sum(at_least_one_counts$USPercent)

# Assuming state_fips contains the state FIPS code and state name mapping
# Merge the "At least one" data with the state FIPS lookup for full state
names
at_least_one_counts <- at_least_one_counts %>%
  left_join(state_fips, by = c("STATE" = "STATE"))

```

GEOCODING BY EXAMPLE WITH BRFSS

```

library(tidygeocoder)
library(tidyverse)
library(survey)

```

```

library(sf)
library(tigris)
library(viridis)

# Create a dataframe with state names
state_fips <- tibble(
  STATE = c(1, 2, 4, 5, 6, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39,
40, 41, 42, 44, 45, 46, 47, 48, 49, 50, 51, 53, 54, 55, 56, 66, 72, 78),
  state = c("Alabama", "Alaska", "Arizona", "Arkansas", "California",
"Colorado", "Connecticut", "Delaware", "District of Columbia", "Florida",
"Georgia", "Hawaii", "Idaho", "Illinois", "Indiana", "Iowa", "Kansas",
"Kentucky", "Louisiana", "Maine", "Maryland", "Massachusetts", "Michigan",
"Minnesota", "Mississippi", "Missouri", "Montana", "Nebraska", "Nevada",
"New Hampshire", "New Jersey", "New Mexico", "New York", "North Carolina",
"North Dakota", "Ohio", "Oklahoma", "Oregon", "Pennsylvania", "Rhode
Island", "South Carolina", "South Dakota", "Tennessee", "Texas", "Utah",
"Vermont", "Virginia", "Washington", "West Virginia", "Wisconsin",
"Wyoming", "Guam", "Puerto Rico", "Virgin Islands")
)

# Geocode the state names to obtain their lat/long
geocoded_states <- state_fips %>%
  geocode(state, method = 'osm', full_results = FALSE)

# Print the geocoded results
print(geocoded_states)

# Merge cancer data with state FIPS lookup and geocoded coordinates
# NEED FROM mapping_geocoding
cancer_by_state_long <- cancer_by_state_long %>%
  left_join(state_fips, by = "state") %>%
  left_join(geocoded_states, by = "state")

# Print the merged data to verify
print(cancer_by_state_long)

# Plot the map for each cancer type category using geocoded coordinates
ggplot(data = cancer_by_state_long) +
  geom_point(aes(x = long, y = lat, color = Proportion, size = Proportion))
+
  scale_color_viridis_c(option = "plasma", na.value = "white") +
  theme_minimal() +
  theme(
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    legend.position = "bottom"
  ) +
  facet_wrap(~Type, ncol = 1) +
  labs(color = "Proportion", size = "Proportion", title = "Proportion of
Cancer Types by State", caption = "Source: BRFSS 2022")

# Load US states shapefile using tigris
us_states <- states(cb = TRUE, resolution = "20m")

# Simplify state names for merging
us_states$NAME <- tolower(us_states$NAME)

```

```

cancer_by_state_long$state <- tolower(cancer_by_state_long$state)

# Define the st_shift function
st_shift <- function(geometry, shift) {
  st_geometry(geometry) <- st_geometry(geometry) + shift
  return(geometry)
}

# Manually adjust Alaska and Hawaii positions
alaska <- us_states %>% filter(NAME == "alaska") %>% st_transform(., crs =
5070) %>% st_shift(., c(2700000, -1500000))
hawaii <- us_states %>% filter(NAME == "hawaii") %>% st_transform(., crs =
5070) %>% st_shift(., c(5200000, -1400000))

# Plot the map for each cancer type category
ggplot(data = us_states) +
  geom_sf(fill = "white", color = "black") + # Plot state boundaries
  geom_point(data = cancer_by_state_long, aes(x = long, y = lat, color =
Proportion, size = Proportion)) + # Plot geocoded points
  scale_color_viridis_c(option = "plasma", na.value = "white") +
  theme_void() +
  theme(
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    legend.position = "bottom"
  ) +
  facet_wrap(~Type, ncol = 2) +
  labs(color = "Proportion", size = "Proportion", title = "Proportion of
Cancer Types by State", caption = "Source: BRFSS 2022")

# Ensure merging dataframes have consistent column names and merge
correctly
us_states <- us_states %>%
  left_join(cancer_by_state_long, by = c("NAME" = "state"))

# Plot the map for each cancer type category with state fill based on
proportion
ggplot(data = us_states) +
  geom_sf(aes(fill = Proportion), color = "black") + # Fill states based
on proportion
  scale_fill_viridis_c(option = "plasma", na.value = "white") +
  theme_minimal() +
  theme(
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    legend.position = "bottom"
  ) +
  facet_wrap(~Type, ncol = 2) +
  labs(fill = "Proportion", title = "Proportion of Cancer Types by State",
caption = "Source: BRFSS 2022")

```

MAPPING BY EXAMPLE WITH BRFSS

```

library(here)
library(tidyverse)
library(viridis)
library(survey)

```

```

library(sf)
library(tigris)

# Reading in the 2022 data
data_2022 <- readRDS(here("temp_data", "2022_data.rds"))
str(data_2022)
colnames(data_2022)

# Removing "-" from all variables - causes problems
data_2022 <- data_2022 %>%
  rename_with(~str_remove(., "_"), starts_with("_"))

# Create a lookup table for state FIPS codes via the 2022 codebook
state_fips <- tibble(
  "STATE" = c(1, 2, 4, 5, 6, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39,
40, 41, 42, 44, 45, 46, 47, 48, 49, 50, 51, 53, 54, 55, 56, 66, 72, 78),
  state = as.factor(c("Alabama", "Alaska", "Arizona", "Arkansas",
"California", "Colorado", "Connecticut", "Delaware", "District of
Columbia", "Florida", "Georgia", "Hawaii", "Idaho", "Illinois", "Indiana",
"Iowa", "Kansas", "Kentucky", "Louisiana", "Maine", "Maryland",
"Massachusetts", "Michigan", "Minnesota", "Mississippi", "Missouri",
"Montana", "Nebraska", "Nevada", "New Hampshire", "New Jersey", "New
Mexico", "New York", "North Carolina", "North Dakota", "Ohio", "Oklahoma",
"Oregon", "Pennsylvania", "Rhode Island", "South Carolina", "South Dakota",
"Tennessee", "Texas", "Utah", "Vermont", "Virginia", "Washington", "West
Virginia", "Wisconsin", "Wyoming", "Guam", "Puerto Rico", "Virgin
Islands")))
)

# Join BRFSS data with state FIPS lookup
data_2022 <- data_2022 %>%
  left_join(state_fips, by = "STATE")

# Levels of state
levels(data_2022$state)

# Create survey design object
data_2022_survey <- svydesign(
  id = ~PSU,
  strata = ~STSTR,
  weights = ~WT2RAKE,
  data = data_2022,
  nest = TRUE
)

# Recode CNCRDIFF as a factor
data_2022$CNCRDIFF <- factor(data_2022$CNCRDIFF, levels = c(1, 2, 3, 7, 9),
  labels = c("One", "Two", "Three or more",
"Don't know/Not Sure", "Refused"))

# Set survey options
options(survey.lonely.psu = "adjust")

# Create survey design object
data_2022_survey_adjusted <- svydesign(
  id = ~PSU,

```

```

    strata = ~STSTR,
    weights = ~WT2RAKE,
    data = data_2022,
    nest = TRUE
  )

# Calculate weighted proportions for the cancer variable by state
cancer_by_state <- svyby(
  ~CNCRDIF,
  ~STATE,
  data_2022_survey_adjusted,
  svymean,
  na.rm = TRUE
)

# Reshape data for easier merging
cancer_by_state_long <- cancer_by_state %>%
  pivot_longer(cols = starts_with("CNCRDIF"), names_to = "Cancer_Type",
    values_to = "Proportion") %>%
  mutate(Type = case_when(
    str_detect(Cancer_Type, "One") ~ "One",
    str_detect(Cancer_Type, "Two") ~ "Two",
    str_detect(Cancer_Type, "Three.or.more") ~ "Three or more",
    str_detect(Cancer_Type, "Don.t.know.Not.Sure") ~ "Don't know/Not Sure",
    str_detect(Cancer_Type, "Refused") ~ "Refused",
    TRUE ~ NA_character_
  )) %>%
  select(-Cancer_Type)

# Merge cancer data with state FIPS lookup
cancer_by_state_long <- cancer_by_state_long %>%
  left_join(state_fips, by = c("STATE" = "STATE"))

# Load US states shapefile using tigris
us_states <- states(cb = TRUE, resolution = "20m")

# Filter out non-contiguous states and territories
us_states <- us_states %>%
  filter(!NAME %in% c("guam", "puerto rico", "virgin islands"))

cancer_by_state_long <- cancer_by_state_long %>%
  filter(!state %in% c("Guam", "Puerto Rico", "Virgin Islands"))

# Merge cancer data with the shapefile
us_states <- us_states %>%
  left_join(cancer_by_state_long, by = c("NAME" = "state"))

# Plot the map for each cancer type category
ggplot(data = us_states) +
  geom_sf(aes(fill = Proportion)) +
  scale_fill_viridis_c(option = "plasma", na.value = "white") +
  theme_void() +
  theme(
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    legend.position = "bottom"
  ) +

```

```

    facet_wrap(~Type, ncol = 2) +
    labs(fill = "Proportion", title = "Number of Cancers by State", caption =
"Source: CDC BRFSS 2022")

```

INTERACTIVE MAPPING BY EXAMPLE WITH BRFSS

```

# Load necessary libraries
library(plotly)
library(sf)
library(rnaturalearth)
library(rnaturalearthdata)
library(tidyverse)
library(PNWCOLORS)

palette <- pnw_palette(name="Shuksan2",n=3,type="continuous")

# Load US states data from rnaturalearth
us_states <- ne_states(country = "united states of america", returnclass =
"sf")

# Merge the "At least one" data with the spatial data
us_states <- us_states %>%
  left_join(at_least_one_counts, by = c("NAME" = "state"))

# Create a hover text column
us_states$hover <- paste0(us_states$name, "\nPercent of U.S.: ",
round(us_states$USPercent, 2))

# Define graph properties
graph_properties <- list(
  scope = 'usa',
  showland = TRUE,
  landcolor = toRGB("white"),
  projection = list(type = 'albers usa')
)

# Define font and label styling for hover
font <- list(
  family = "DM Sans",
  size = 15,
  color = "black"
)

label <- list(
  bgcolor = "#EEEEEE",
  bordercolor = "transparent",
  font = font
)

# Create the Plotly choropleth map for the "At least one" category
proportion
cancer_graph <- plot_geo(us_states, locationmode = "USA-states") %>%
  add_trace(
    locations = ~postal, # Use the postal column for state
    codes
    z = ~USPercent,
    zmin = 0,

```

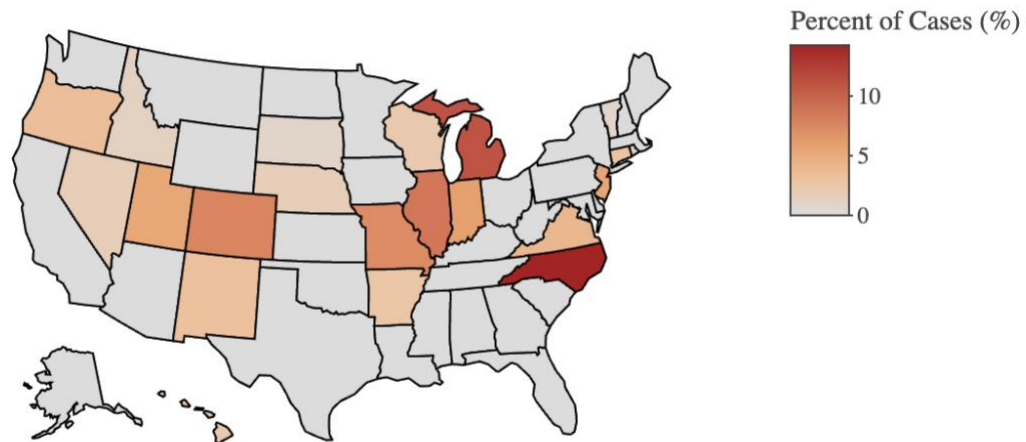
```

    zmax = max(us_states$USPercent, na.rm = TRUE),
    color = ~USPercent,
    colorscale = palette,          # Use custom color scale
    text = ~hover,
    hoverinfo = 'text'
) %>%
layout(
  geo = graph_properties,
  title = list(
    text = "Percent of U.S. At Least One Cancer Case by State",
    font = list(family = "DM Sans", size = 24) # Increase title font
size
  ),
  font = list(family = "DM Sans"),
  annotations = list(
    list(
      x = 0.5,
      y = -0.1, # Adjust this value as needed to position the footnote
      text = "Source: BRFSS 2022. Note: Data represents the proportion of
at least one cancer case.",
      showarrow = FALSE,
      xref = "paper",
      yref = "paper",
      xanchor = "center",
      yanchor = "top",
      font = list(size = 12)
    )
  )
) %>%
config(displayModeBar = FALSE) %>%
style(hoverlabel = label) %>%
colorbar(
  title = "Percent of Cases (%)",          # Rename legend title
  tickprefix = ''
)

cancer_graph

```

Percent of U.S. At Least One Cancer Case by State



Source: BRFSS 2022. Note: Data represents the proportion of at least one cancer case.

COMPARISON OF TECHNIQUES

ADVANTAGES AND DISADVANTAGES CHART

TBD

USE CASES OF SAS

TBD

USE CASES OF R

TBD

CONCLUSION

TBD - note that both SAS and R are constantly evolving and improving visualization capability, so it is important to keep up with developments in whichever package is being used.

REFERENCES

TBD - all the papers I put in the shared folder can be cited.

ACKNOWLEDGMENTS

Joshua J. Cook thanks his mentors, Louise Hadden, Dr. Achraf Cohen, Dr. Swann Adams, and Kirk Paul Lafler, as well as Richann Watson and Troy Martin Hughes for their general guidance and mentorship.

RECOMMENDED READING

- TBD

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Joshua J. Cook, M.S. DS, M.S. CRM, ACRP-PM, CCRC
Adjunct Professor / Clinical Researcher / Data Scientist
Website: <https://joshua-j-cook-portfolios.netlify.app/>
Email: jcook0312@outlook.com

—

Louise S. Hadden
Abt Associates, Inc.
Email: louise_hadden@abtassoc.com

—

Swann A. Adams, Ph.D, M.S., FACE
Professor
Department of Biobehavioral Health and Nursing Science
College of Nursing and the Department of Epidemiology & Biostatistics
Arnold School of Public Health
Email: swann.adams@sc.edu

Any brand and product names are trademarks of their respective companies.