Tracking Counties Visualizations

Evan Canfield

5/11/2019

#Purpose The purpose of this notebook is to visualize the tracking of states and counties visited.

#Setup The following libraries were used.

Libaries

```
#Library Calls

# Install pacman if not available
if (!require(pacman)) {install.packages('pacman')}
library(pacman)

# Install and load the following libraries
pacman::p_load(
  broom,
  glue,
  grid,
  gridExtra,
  ggthemes,
  here,
  leaflet,
  RColorBrewer,
  sf,
  tidyverse
  )

# Define Global Settings
# Project root directory
ROOT_DIR = here::here()

# Year (For Shapefile Import)
YEAR <- 2022
```

##Inputs ###Shapefiles ####States

```
file_name <- glue::glue("cb_{YEAR}_us_state_500k.shp")
file_path <- file.path(ROOT_DIR, "data", "shapefiles", "state", YEAR, file_name)

state.sf <- sf::read_sf(dsn = file_path)
```

####County

```
file_name <- glue::glue("cb_{YEAR}_us_county_500k.shp")
file_path <- file.path(ROOT_DIR, "data", "shapefiles", "county", YEAR, file_name)
```

```
county.sf <- sf::read_sf(dsn = file_path)
```

###Tracking Table

```
file_path <- file.path(ROOT_DIR, "data", "tables", "list_of_counties_active.csv")

tracking_counties.df = readr::read_csv(
  file = file_path,
  col_types = readr::cols(
    visited = readr::col_integer(),
    date = readr::col_datetime(format = "%Y/%m/%d/")
    )
  )
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

#Processing ##Create GEOID

```
tracking_counties.df <- tracking_counties.df %>%
  dplyr::mutate(
    GEOID = stringr::str_c(state_code, county_code)
  )
```

##Create State Tracking Table

```
tracking_states.df <- tracking_counties.df %>%
  dplyr::group_by(state_code) %>%
  dplyr::mutate(
    n_counties = sum(visited),
    visited = dplyr::if_else(n_counties > 0, 1, 0),
  ) %>%
  dplyr::distinct(state, state_code, state_name, visited)
```

##Join Tracking Tables ###States

```
tracking_states_join.df <- tracking_states.df %>%
  dplyr::select(state_code, visited)

state_tracking.sf <- state.sf %>%
  dplyr::left_join(
    tracking_states_join.df,
    by = c("STATEFP" = "state_code")
  )
```

###Counties

```
tracking_counties_join.df <- tracking_counties.df %>%
  dplyr::rename(NAME_STATE = state_name) %>%
  dplyr::select(GEOID, NAME_STATE, visited, date)

county_tracking.sf <- county.sf %>%
  dplyr::left_join(
    tracking_counties_join.df,
    by = "GEOID"
```

```
  ) %>%
  dplyr::rename(NAME_COUNTY = NAME)
```

Plots

Shared Properties

```
cols <- c("#F4F6F6", "#1F618D")
```

Contiguous US

```
# Define states and territory codes to drop
states_terr_to_drop <- c("02", "15", "60", "66", "69", "72", "78")
```

States

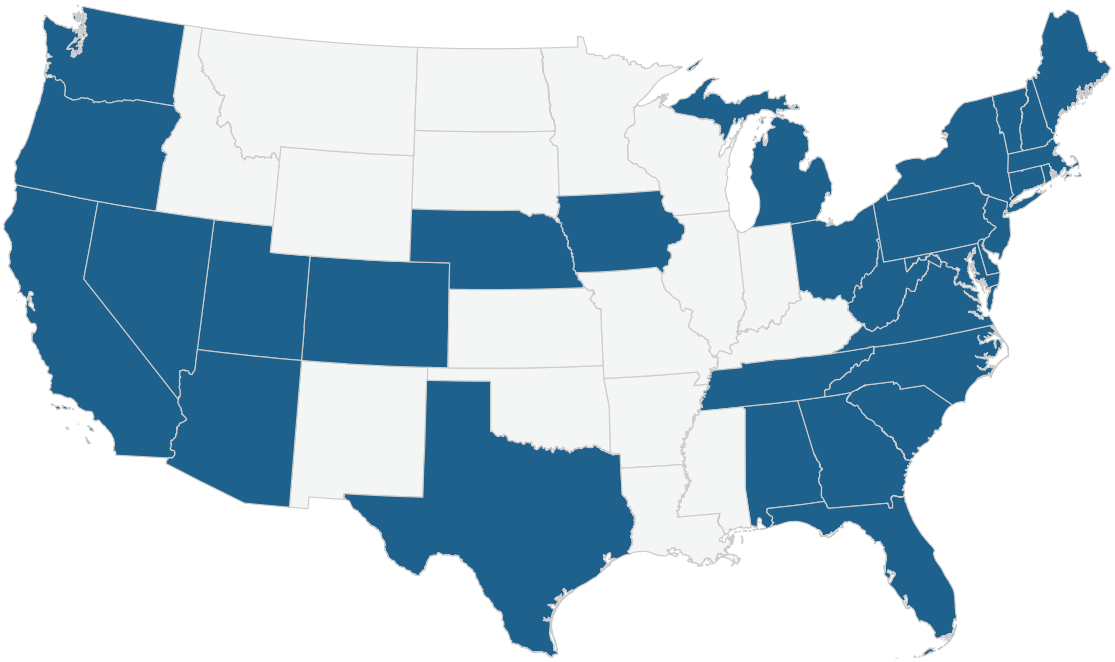
```
# Drop non-contiguous states and territories
plot.sf <- state_tracking.sf %>%
 dplyr::filter(!(STATEFP %in% states_terr_to_drop))

# Change projection
plot_reprojected_state.sf <- sf::st_transform(plot.sf, 'EPSG:3082')

p <- ggplot(
  data = plot_reprojected_state.sf,
  mapping = aes(fill = factor(visited))
  ) +
  geom_sf(color = "gray80", size = 0.05) +
  scale_fill_manual(values = cols) +
  theme_map() +
  guides(fill="none") +
  labs(title = "United States")

p
```

United States



Counties

```
# Drop non-contiguous states and territories
plot.sf <- county_tracking.sf %>%
 dplyr::filter(!(STATEFP %in% states_terr_to_drop))

# Change projection
plot_reprojected_county.sf <- sf::st_transform(plot.sf, 'EPSG:3082')

p <- ggplot() +
  geom_sf(
    data = plot_reprojected_county.sf,
    mapping = aes(fill = factor(visited)),
    color = "gray80", size = 0.05
    ) +
  geom_sf(
    data = plot_reprojected_state.sf,
    mapping = aes(
      fill = factor(visited),
      alpha = factor(visited)
      ),
    # alpha = 0.1,
    color = "black",
    size = 0.25
    ) +
  scale_fill_manual(values = cols) +
  scale_alpha_manual(values = c(0, 0.1)) +
  theme_map() +
  guides(
    fill="none",
```

```
    alpha = "none"
    ) +
  labs(title = "United States")

p
```

United States