Template to make PSRC leaflet map

PSRC Data Science

2023-05

How to quickly make a formatted map

This template shows you how to make a map given a numeric or categorical data item with a geographic id and a shape file with the same ID.

Inputs required: a data table with a field to map and an ID, a geographic layer with a matching ID

User options: Legend tile, Legend subtitle color ramp

0. Clone the repository https://github.com/psrc/psrc.map

and set working directory to directory with code map_region.R

```
setwd("C:/GitHub/psrc.map/")
```

1. Read in mapping and helper libraries

```
library(sf) # geographic data library
library(dplyr) # for working with data frames
library(psrcelmer) # to get data from Elmer
library(psrcplot) # to get color schemes
library(mapview) # for outputting the map to html or png
```

2. Read in any other libraries or keys you need

```
library(psrccensus)
Sys.getenv("CENSUS_API_KEY")
```

3. Define create_psrc_map function

This **map_region.R** file has a function called **create_psrc_map** to create a leaflet map for you

```
# should be in your working directory
source('map_region.R')
```

4. Read in Data Table with a geographic field ID

Prepare data to join with the geographic layer Note: you will need field names to match on This example gets ACS 5-year data for Asian Groups by Tract.

```
big_tbl <-psrccensus::get_acs_recs(geography='tract',table.names='B02015',year=2021, acs.type='acs5')
```

5. Clean up the data in preparation to join to the geographic layer

```
# read in data table and process data to join and map
tbl <- big_tbl %>%filter(label=='Estimate!!Total:!!Vietnamese')%>%
  dplyr::select(GEOID,estimate) %>%
  dplyr::mutate(dplyr::across(c('GEOID'), as.character))%>%
  dplyr::group_by(GEOID) %>%
  dplyr::summarise(Total=sum(estimate))
```

6. Read in ElmerGeo geographic layer

```
tract_layer_name <- "TRACT2020_NOWATER"
lyr <- st_read_elmergeo(tract_layer_name)
```

7. Join to data table layer via matching IDs, choose the field to map

```
lyr_data<- dplyr::left_join(lyr,tbl, by = c("geoid20"="GEOID"))
# this is the field to map
lyr_data_field<-lyr_data$Total
```

8. Call the function to make the map.

these are the function inputs:

lyr: sf map layer, the geographic layer including the field you want to map

lyr_data_field: numeric vector of data values to map

legend_title: string, The title in the legend

legend_subtitle: string, The title in the legend

psrc_col_pal: string list, color ramp

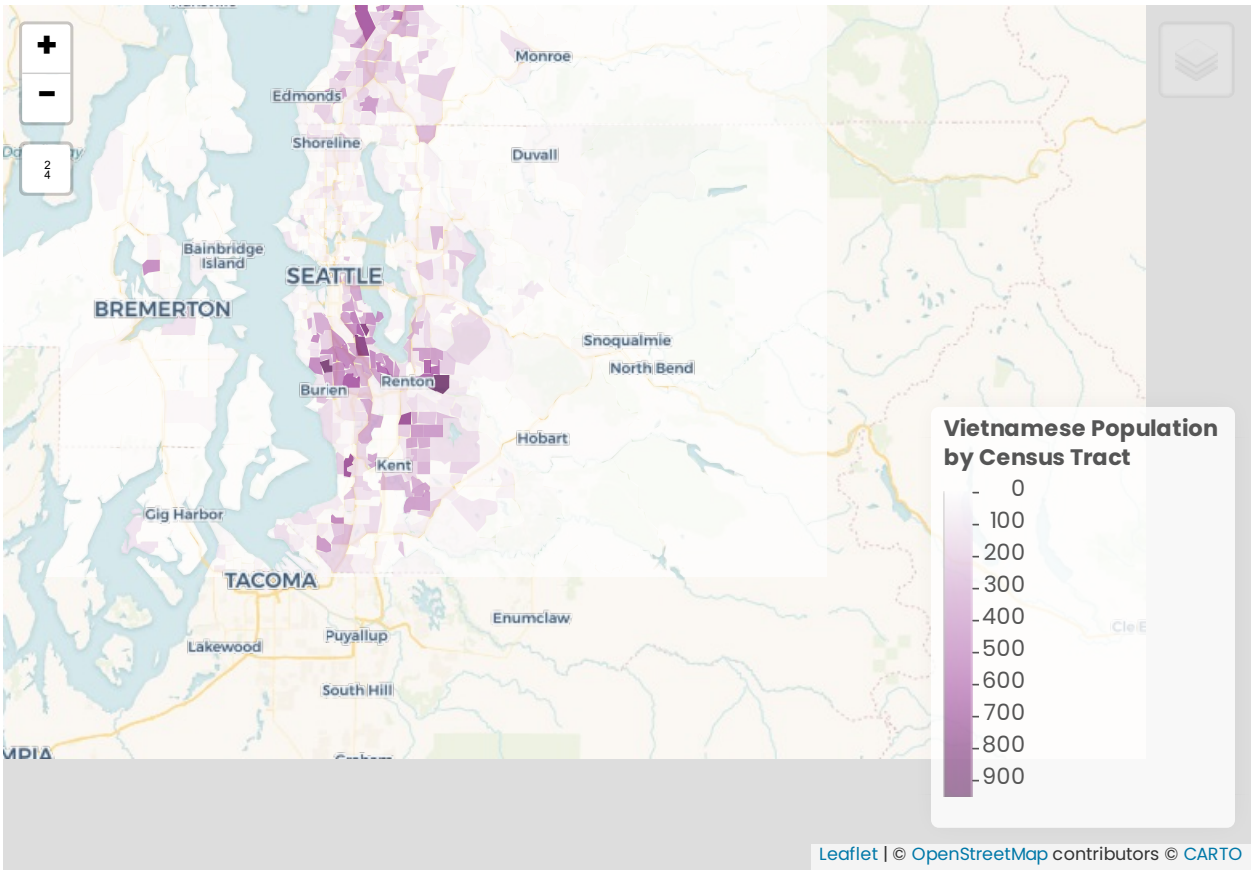
map_lat=47.615, map_lon=-122.257: defaults, can be overridden for centering the map

map_zoom=8.5, wgs84=4326 : defaults, can be overriden, zoom level and projection

```
# need to add a white color to ensure contrast
psrc_purple_plus<-append(psrc_colors$purples_inc, "#FFFFFF", after=0)
map_it<-create_psrc_map(lyr=lyr_data,
                        lyr_data_field=lyr_data_field,
                        legend_title= 'Vietnamese Population',
```

```
                        legend_subtitle = 'by Census Tract',
                        psrc_col_pal=psrc_purple_plus)

map_it
```



Step 9. (optional) Save the map as html

```
mapshot(map_it, url= "VietnamesePop.html")
```

Step 10. (optional) Save the map as png

```
mapshot(map_it, file= "VietnamesePop.png")
```