

DEM 5093/7093 Lab 1 using R - Basic Map making

Julia Kay Wolf, Ph.D.

2023-01-27

Get a Census developer API Key

Obtain one at the Census Developer website

Save your API key to your working directory

```
use_tidycensus::census_api_key(key = "yourkeyhere", install = T)
```

One time to install your key for use in tidycensus

```
library(tidycensus)
library(tidyverse)
library(sf)
library(ggplot2)
library(classInt)
library(dplyr)
```

Examine data profile tables

The `load_variables()` function will load all available variables in the ACS for a given year

```
v15_Profile <- load_variables(year = 2019 ,
                             dataset = "acs5/profile",
                             cache = TRUE)
```

Calling `View(v15_Profile)` will let you interactively browse and filter the ACS variables, this is one way to search for what you're looking for.

Search for variables by keywords in the label

```
v15_Profile%>%
  filter(grepl(pattern = "POVERTY", x = label))%>%
  select(name, label)
```

```
## # A tibble: 38 x 2
##   name      label
##   <chr>    <chr>
```

```
## 1 DP03_0119 Estimate!!PERCENTAGE OF FAMILIES AND PEOPLE WHOSE INCOME IN THE P~
## 2 DP03_0119P Percent!!PERCENTAGE OF FAMILIES AND PEOPLE WHOSE INCOME IN THE PA~
## 3 DP03_0120 Estimate!!PERCENTAGE OF FAMILIES AND PEOPLE WHOSE INCOME IN THE P~
## 4 DP03_0120P Percent!!PERCENTAGE OF FAMILIES AND PEOPLE WHOSE INCOME IN THE PA~
## 5 DP03_0121 Estimate!!PERCENTAGE OF FAMILIES AND PEOPLE WHOSE INCOME IN THE P~
## 6 DP03_0121P Percent!!PERCENTAGE OF FAMILIES AND PEOPLE WHOSE INCOME IN THE PA~
## 7 DP03_0122 Estimate!!PERCENTAGE OF FAMILIES AND PEOPLE WHOSE INCOME IN THE P~
## 8 DP03_0122P Percent!!PERCENTAGE OF FAMILIES AND PEOPLE WHOSE INCOME IN THE PA~
## 9 DP03_0123 Estimate!!PERCENTAGE OF FAMILIES AND PEOPLE WHOSE INCOME IN THE P~
## 10 DP03_0123P Percent!!PERCENTAGE OF FAMILIES AND PEOPLE WHOSE INCOME IN THE PA~
## # ... with 28 more rows
```

```
v15_Profile%>%
  filter(grepl(pattern = "Built 2000 to 2009", x = label))%>%
  select(name, label)
```

```
## # A tibble: 2 x 2
##   name      label
##   <chr>     <chr>
## 1 DP04_0019 Estimate!!YEAR STRUCTURE BUILT!!Total housing units!!Built 2000 to~
## 2 DP04_0019P Percent!!YEAR STRUCTURE BUILT!!Total housing units!!Built 2000 to ~
```

Extract from ACS summary file data

The `tidycensus` package has a function `get_acs()` that will download data from the Census API for you automatically **assuming you've installed your key from above**

Here I get data profile variables from 2017 for Bexar County, TX Census Tracts

Here is a query where we extract several variables from the 2017 ACS for Bexar County, Texas. We can also get the spatial data by requesting `geometry=TRUE`.

Using `output="wide"` will put each variable in a column of the data set, with each row being a census tract.

```
sa_acs<-get_acs(geography = "tract",
                state="TX",
                county = "Bexar",
                year = 2019,
                variables=c("DP05_0001E", "DP03_0119PE") ,
                geometry = T,
                output = "wide")
```

```
## Getting data from the 2015-2019 5-year ACS
```

```
## Downloading feature geometry from the Census website. To cache shapefiles for use in future sessions.
```

Using the ACS Data Profile

```
head(sa_acs)
```

Here, I create some other variables that we may need later

```
# create a county FIPS code - 5 digit
sa_acs$county<-substr(sa_acs$GEOID, 1, 5)
# rename variables and filter missing cases
sa_acs2<-sa_acs%>%
  mutate(totpop= DP05_0001E,
         ppov=DP03_0119PE) %>%
  st_transform(crs = 2919)%>%
  na.omit()
```

Write data out to shapefile

You may need to create or modify some data in R and then use it in the desktop GIS (QGIS), we can write any data from R into a variety of data formats using the `sf::st_write()` function.

```
#eval=F means eval=FALSE which means this code is not running, it is here for you to view though  
st_geometry(sa_acs2)<-NULL  
foreign::write.dbf(as.data.frame(sa_acs2), file="C:/Users/xee291/OneDrive - University of Texas at San Antonio/Desktop/UTSA 2022-2023/UTSA_2022-2023.shp.dbf")  
  
#change the directory for your computer  
sf::st_write(sa_acs2,  
             dsn="C:/Users/xee291/OneDrive - University of Texas at San Antonio/Documents/UTSA 2022-2023/UTSA_2022-2023.shp",  
             layer="sa_tr_dp03",  
             driver="GPKG") ## Save as geopackage format - QGIS likes this
```

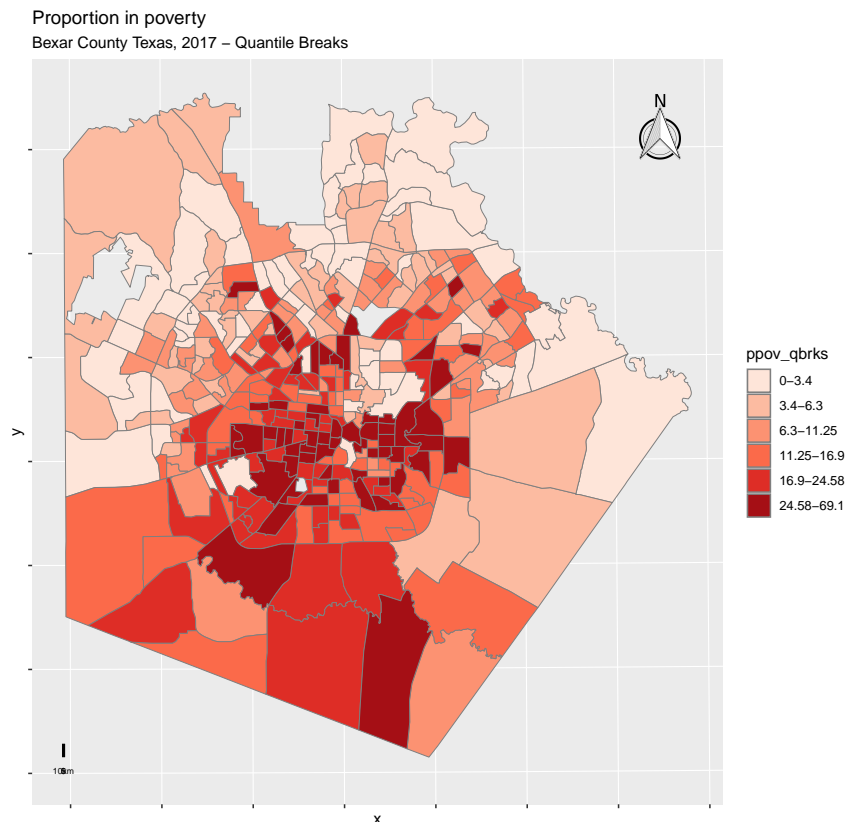
```
mydat<- st_read("C:/Users/xee291/OneDrive - University of Texas at San Antonio/Documents/UTSA 2022-2023/  
names(mydat)
```

Some basic mapping of variables

Here I generate a quantile break for % black in census tracts and compare it to a Jenks break. *Note* in ggplot, the Jenks break is harder to do

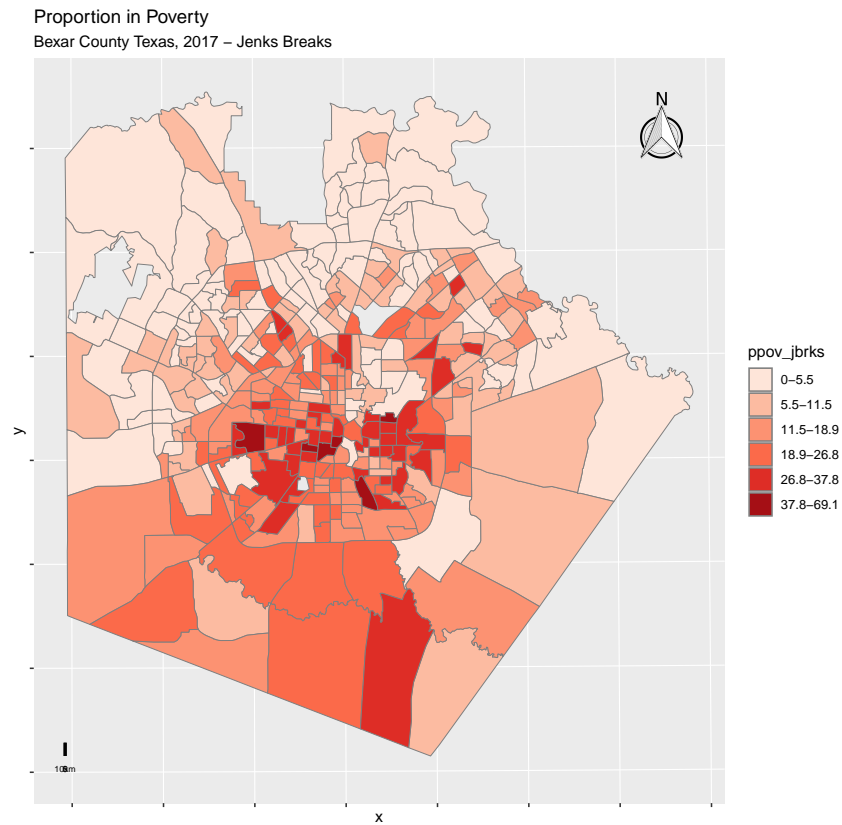
```
library(patchwork)
library(ggsn)
library(xplorerr)
source("https://raw.githubusercontent.com/coreysparks/Rcode/master/mutate_map_funs.R")
ppov_map<-sa_acs2 %>%
  mutate_map_brks(ppov, n=6, style="quantile")%>%
  mutate_map_brks(ppov, n=6, style="jenks")
p1<-ggplot(ppov_map, aes(fill = ppov_qbrks)) +
  geom_sf(color="grey50") +
  ggtitle("Proportion in poverty",
    subtitle = "Bexar County Texas, 2017 - Quantile Breaks")+
  scale_fill_brewer(palette = "Reds") +
  scale_color_brewer(palette = "Reds")+
  theme(axis.text.x = element_blank(),
    axis.text.y = element_blank())+
  north(ppov_map)+
  scalebar(ppov_map, location="bottomleft",
    dist=5, transform = T,
    dist_unit = "km",
    model="WGS84",
    st.size =2 )
```

p1

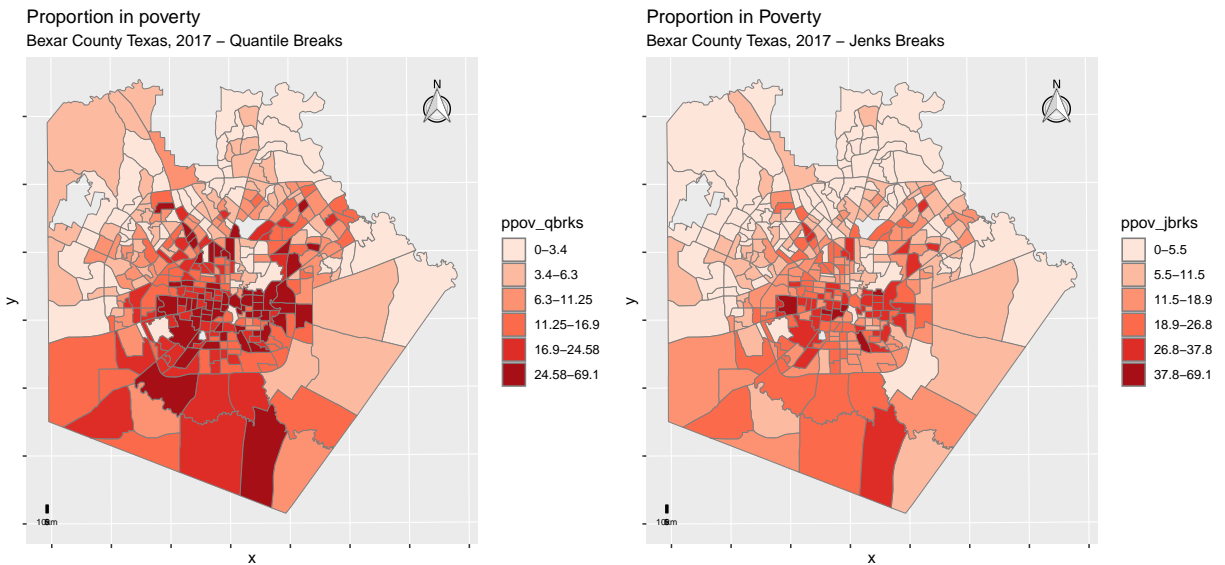


```
p2<-ggplot(ppov_map, aes(fill = ppov_jbrks)) +
  geom_sf(color="grey50") +
  ggtitle("Proportion in Poverty",
    subtitle = "Bexar County Texas, 2017 - Jenks Breaks")+
  scale_fill_brewer(palette = "Reds") +
  scale_color_brewer(palette = "Reds")+
  theme(axis.text.x = element_blank(),
    axis.text.y = element_blank())+
  north(ppov_map)+
  scalebar(ppov_map,
    location="bottomleft",
    dist=5,
    transform = T,
    dist_unit = "km",
    model="WGS84",
    st.size =2)
```

p2



p1 + p2



You can save the image from above to your computer by using `ggsave()`

```
ggsave(filename="C:/Users/xee291/OneDrive - University of Texas at San Antonio/Documents/UTSA 2022-2023",
        dpi = "print")
```

Saving 6.5 x 4.5 in image

Slightly easier way using tmap

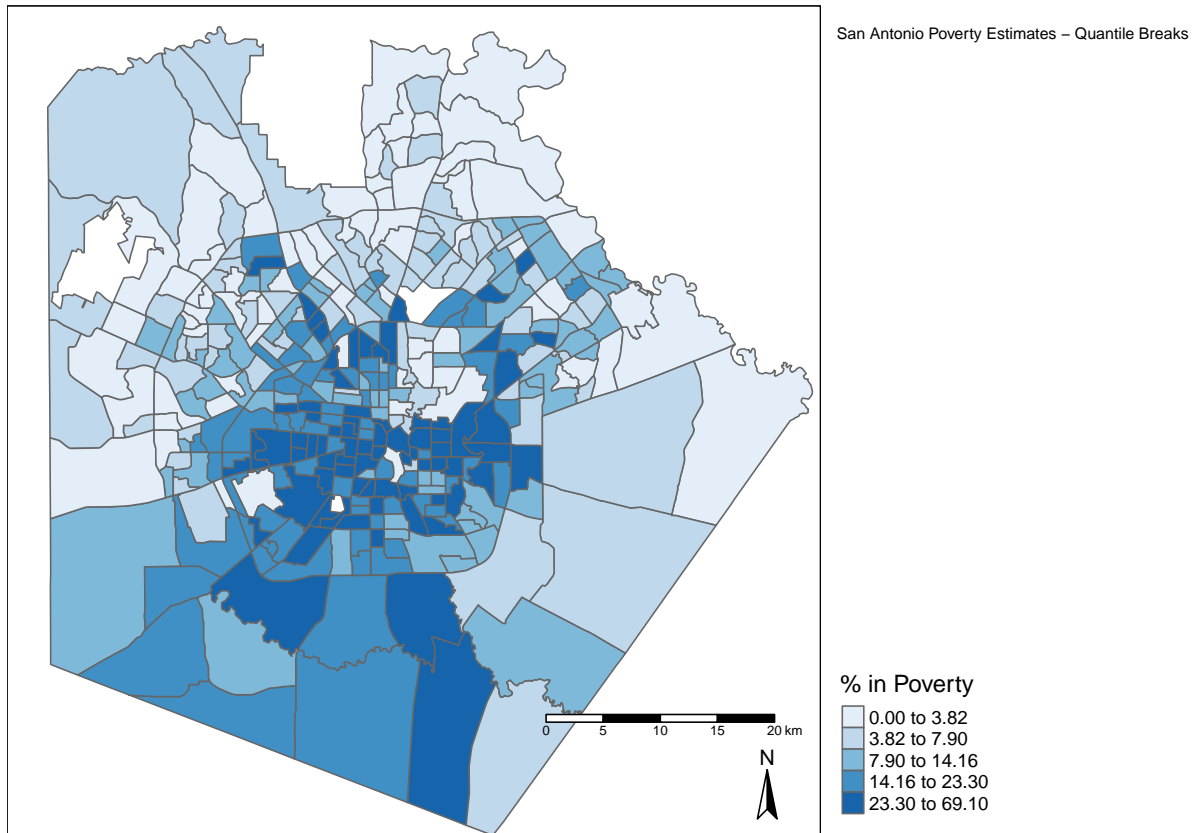
The `tmap` package is an attractive alternative to using `ggplot()` when making maps, and makes basic cartographic principles easier.

Note: `style="fisher"` is equivalent to `style="jenks"` and scales better to larger data.

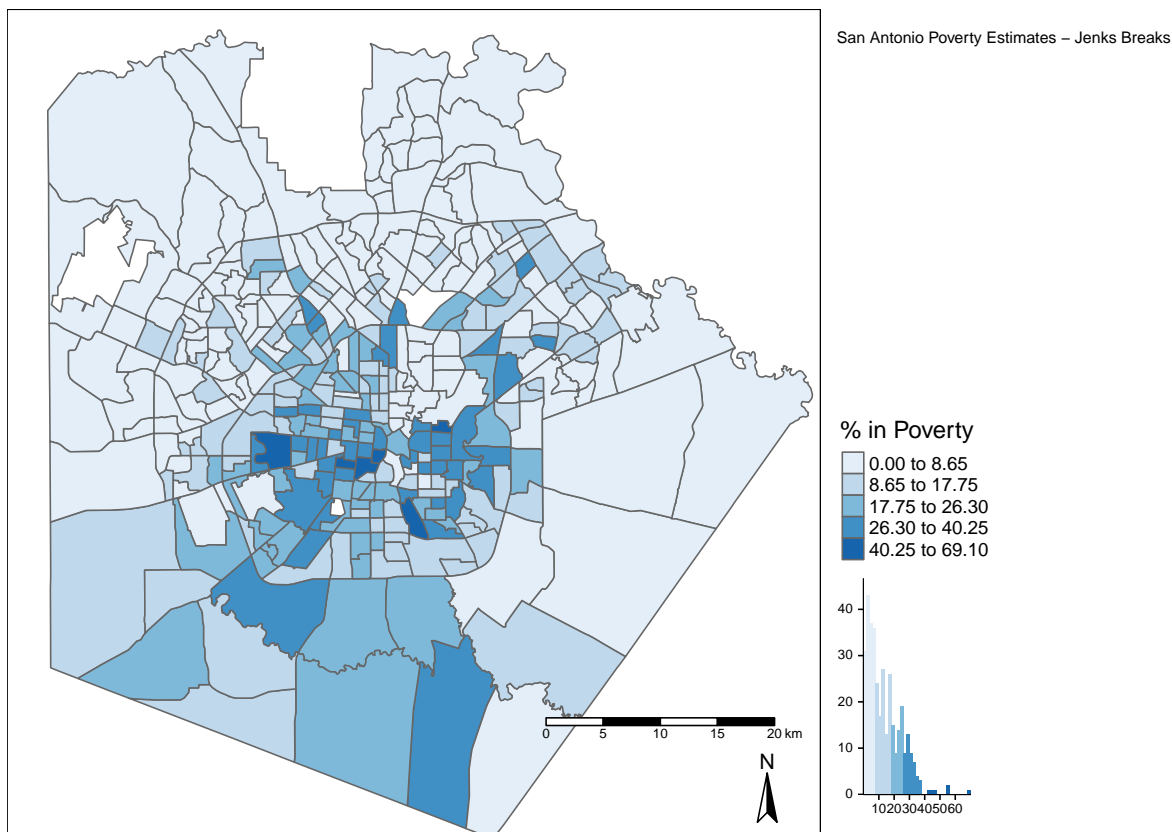
Note: View this website for more `style` and other `tmap` tricks: https://bookdown.org/nicohahn/making_maps_with_r5/docs/tmap.html

```
library(tmap)
library(tmapttools)
tm_shape(sa_acs2)+
  tm_polygons("ppov",
              title="% in Poverty",
              palette="Blues",
              style="quantile", n=5 )+
  tm_format("World",
            title="San Antonio Poverty Estimates - Quantile Breaks",
            legend.outside=T)+
```

```
tm_scale_bar()+
tm_compass()
```



```
tm_shape(sa_acs2)+
  tm_polygons("ppov",
    title="% in Poverty",
    palette="Blues",
    style="fisher",
    n=5,
    legend.hist=T )+
  tm_format("World",
    title="San Antonio Poverty Estimates - Jenks Breaks",
    legend.outside=T)+
  tm_scale_bar()+
  tm_compass()
```

Interactive map with mapview

(Not necessary for Homework 2, but here to view)

```
library(mapview)
library(RColorBrewer)
ppov_map$ppov_jbrks<-relevel(ppov_map$ppov_jbrks,ref = "0-5.5" )
pal <- colorRampPalette(brewer.pal(7, "Blues")) #set colors
mapview(ppov_map,
        zcol="ppov_jbrks",
        legend=T,
        map.types="OpenStreetMap",
        layer.name="% in Poverty")
```

PhantomJS not found. You can install it with `webshot::install_phantomjs()`. If it is installed, please