

king_dav_ps04

Dav King

3/23/2022

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr    0.3.4
## v tibble   3.1.6     v dplyr    1.0.7
## v tidyverse 1.1.4     v stringr  1.4.0
## v readr    2.1.1     vforcats  0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()

library(socviz)
library(tidycensus)
library(tigris)

## Warning: package 'tigris' was built under R version 4.1.3

## To enable caching of data, set 'options(tigris_use_cache = TRUE)'
## in your R script or .Rprofile.

##
## Attaching package: 'tigris'

## The following object is masked from 'package:tidycensus':
## 
##     fips_codes

library(viridis)

## Loading required package: viridisLite

options(tigris_use_cache = T)
```

```

pop_names <- tribble(
  ~varname, ~clean_name,
  "B01003_001", "pop",
  "B01001B_001", "black",
  "B01001A_001", "white",
  "B01001H_001", "nh_white",
  "B01001I_001", "hispanic",
  "B01001D_001", "asian"
)

pop_names

## # A tibble: 6 x 2
##   varname     clean_name
##   <chr>       <chr>
## 1 B01003_001  pop
## 2 B01001B_001 black
## 3 B01001A_001 white
## 4 B01001H_001 nh_white
## 5 B01001I_001 hispanic
## 6 B01001D_001 asian

inc_names <- tribble(
  ~varname, ~clean_name,
  "S1901_C01_012", "median_hh_inc")

fips_pop <- get_acs(geography = "county",
                     variables = pop_names$varname,
                     cache_table = TRUE) %>%
  mutate(variable = reduce2(pop_names$varname,
                           pop_names$clean_name,
                           str_replace,
                           .init = variable)) %>%
  select(-moe) %>%
  pivot_wider(names_from = variable, values_from = estimate) %>%
  rename(fips = GEOID, name = NAME)

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

fips_inc <- get_acs(geography = "county",
                     variables = inc_names$varname,
                     cache_table = TRUE) %>%
  mutate(variable = str_replace(variable,
                               inc_names$varname,
                               inc_names$clean_name)) %>%
  rename(fips = GEOID, name = NAME)

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

## Using the ACS Subject Tables

```

```
fips_map <- get_acs(geography = "county",
                     variables = "B01001_001",
                     geometry = TRUE,
                     shift_geo = FALSE,
                     cache_table = TRUE) %>%
  select(GEOID, NAME, geometry) %>%
  rename(fips = GEOID, name = NAME)
```

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

Questions

Question 1

```
fips_prop <- fips_pop %>%
  pivot_longer(cols = white:hispanic, names_to = "race",
               values_to = "racepop") %>%
  group_by(name) %>%
  summarize(race, prop = (racepop/pop)*100000) %>%
  pivot_wider(names_from = race, values_from = prop)
```

```
## ‘summarise()’ has grouped output by ‘name’. You can override using the ‘.groups’ argument.
```

```
fips_prop
```

```
## # A tibble: 3,220 x 6
## # Groups:   name [3,220]
##   name           white   black   asian nh_white hispanic
##   <chr>        <dbl>   <dbl>   <dbl>    <dbl>    <dbl>
## 1 Abbeville County, South Carolina 69927.  27628.  321.    68851.   1490.
## 2 Acadia Parish, Louisiana       79344.  17358.  88.1    77380.   2554.
## 3 Accomack County, Virginia     68197.  28559.  811.    60028.   8965.
## 4 Ada County, Idaho            90478.  1373.   2543.    84664.   8202.
## 5 Adair County, Iowa          97544.   550.   677.    95992.   2061.
## 6 Adair County, Kentucky      94886.  3137.   624.    93279.   1956.
## 7 Adair County, Missouri      91608.  2680.   2515.    89826.   2452.
## 8 Adair County, Oklahoma      42124.  162.    801.    39892.   6832.
## 9 Adams County, Colorado      81560.  3375.   3954.    50023.   40028.
## 10 Adams County, Idaho       95460.  391.     0       92116.   3588.
## # ... with 3,210 more rows
```

Question 2

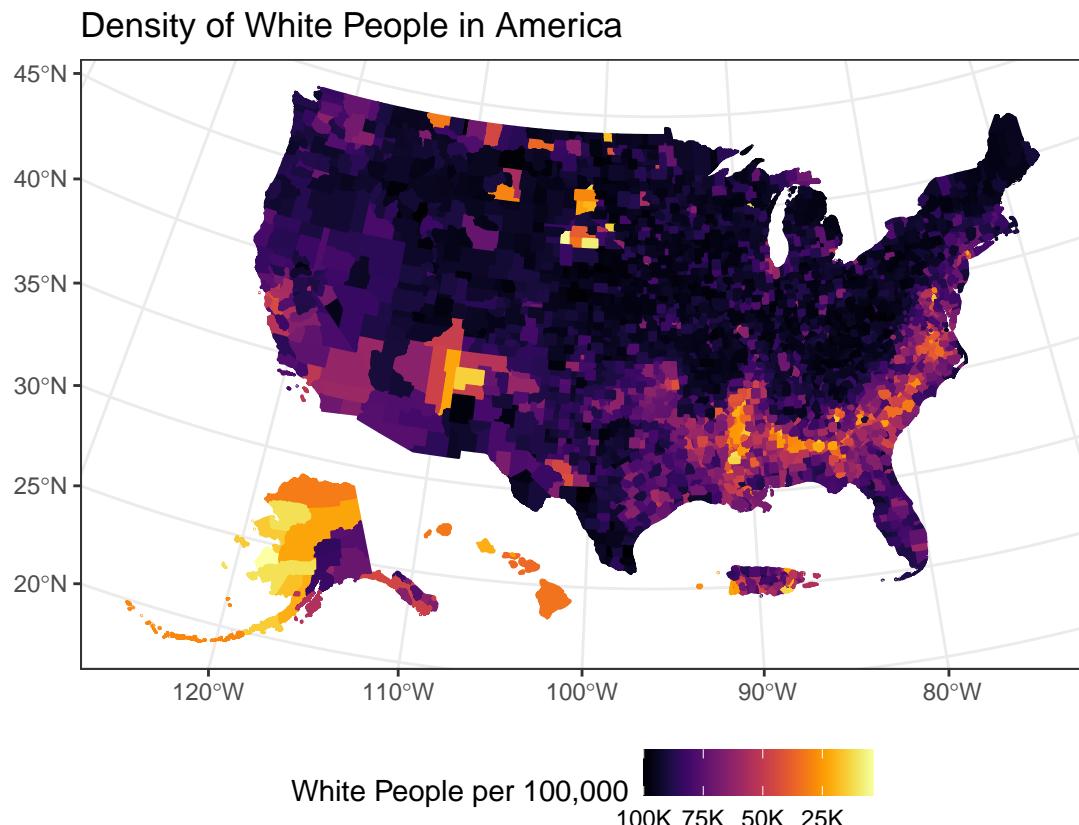
```
full <- left_join(fips_map, fips_prop) %>%
  shift_geometry()
```

```
## Joining, by = "name"
```

We know it works because our output is a data frame with the exact same number of observations but one more row, and looking at it we can see that the one additional row is geometry.

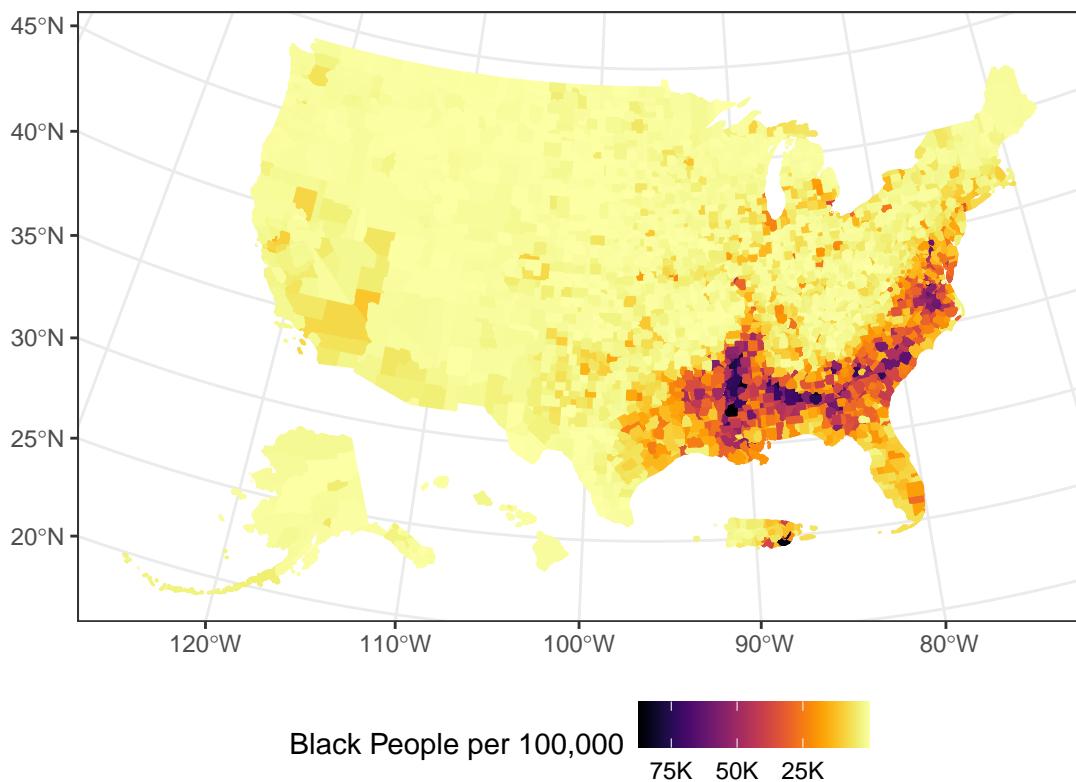
Question 3

```
race_map <- function(race){ggplot(full, aes(fill = race, color = race)) +  
  geom_sf() +  
  theme_bw() +  
  scale_fill_viridis(trans = "reverse", option = "B",  
                     breaks = c(25000, 50000, 75000, 100000),  
                     labels = c("25K", "50K", "75K", "100K")) +  
  scale_color_viridis(trans = "reverse", option = "B") +  
  theme(legend.position = "bottom") +  
  guides(color = "none")  
}  
  
race_map(full$white) +  
  labs(title = "Density of White People in America",  
       fill = "White People per 100,000")
```



```
race_map(full$black) +  
  labs(title = "Density of Black People in America",  
       fill = "Black People per 100,000")
```

Density of Black People in America

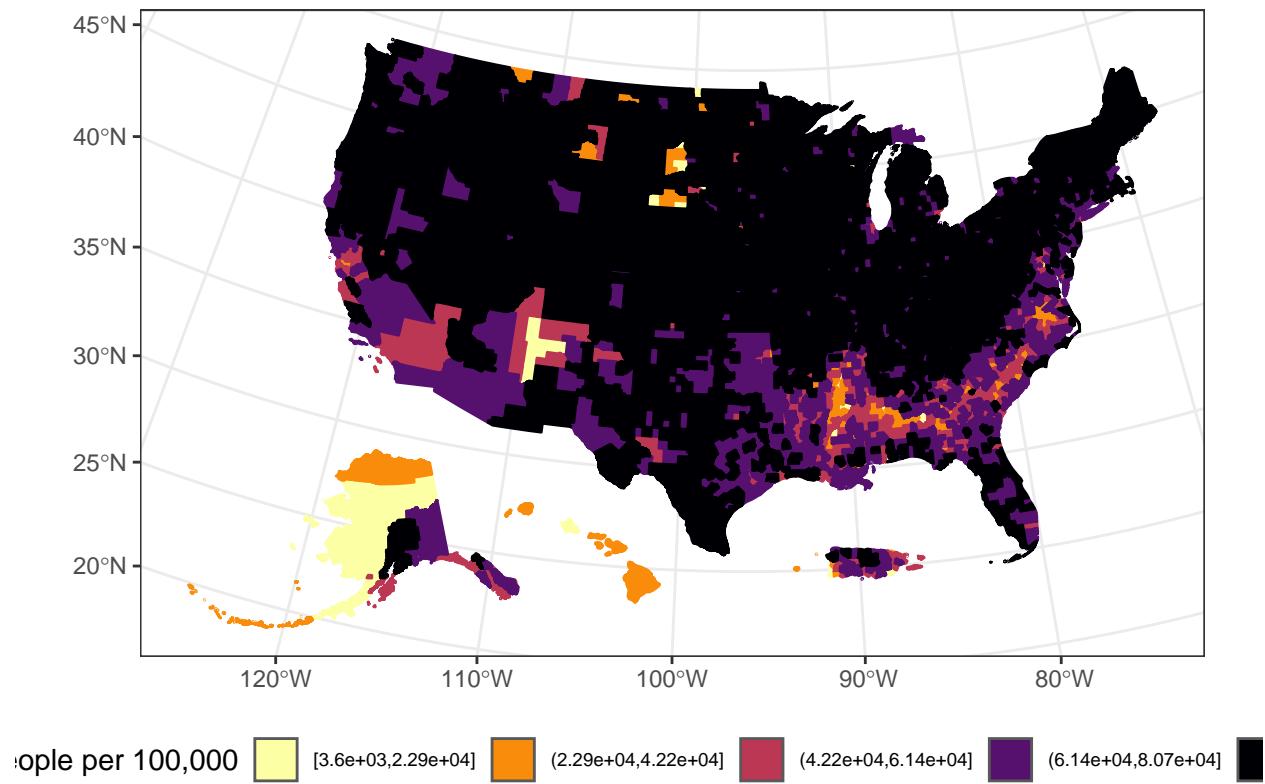


Question 4

```
binned_race_map <- function(race){full %>%
  count(cut = cut_interval(race, 5)) %>%
  ggplot(aes(fill = factor(cut), color = factor(cut))) +
  geom_sf() +
  theme_bw() +
  scale_fill_viridis(discrete = T, option = "B", direction = -1) +
  scale_color_viridis(discrete = T, option = "B", direction = -1) +
  guides(color = "none") +
  theme(legend.position = "bottom",
        legend.text = element_text(size = 6.5))
}

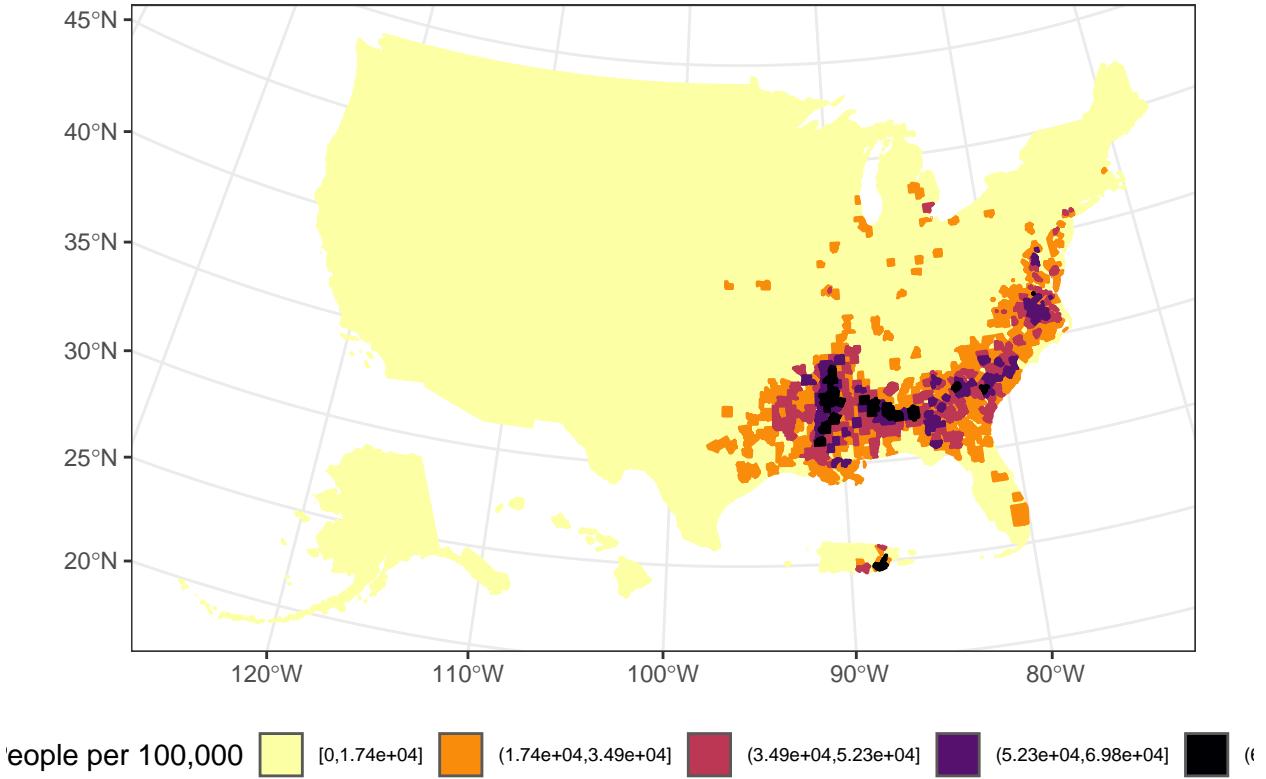
binned_race_map(full$white) +
  labs(title = "Density of White People in America",
       fill = "White People per 100,000")
```

Density of White People in America



```
binned_race_map(full$black) +  
  labs(title = "Density of Black People in America",  
       fill = "Black People per 100,000")
```

Density of Black People in America



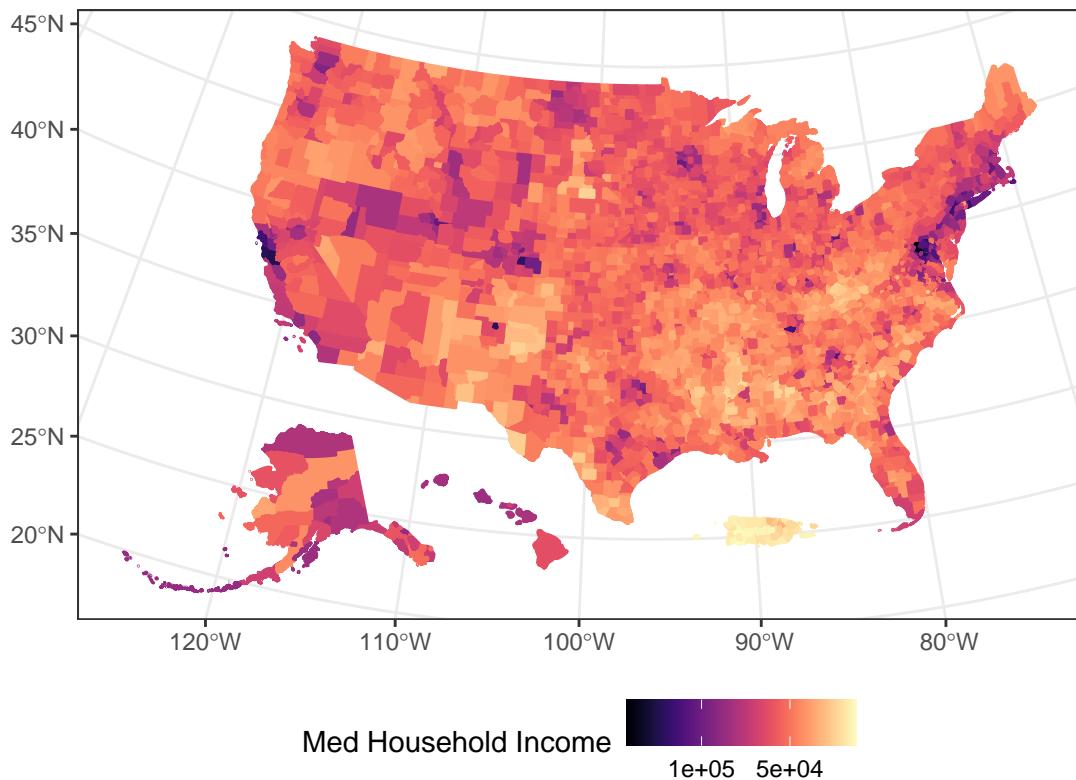
I generally consider this to be a more informative map - since the human eye is incapable of properly integrating across the entire continuous scale, even though this has less actual data on it it remains a more informative, big-picture map.

Question 5

```
fips_map %>%
  left_join(fips_inc) %>%
  shift_geometry() %>%
  ggplot(aes(fill = estimate, color = estimate)) +
  geom_sf() +
  theme_bw() +
  scale_fill_viridis(trans = "reverse", option = "A") +
  scale_color_viridis(trans = "reverse", option = "A") +
  theme(legend.position = "bottom") +
  labs(title = "Median Household Income in America",
       fill = "Med Household Income") +
  guides(color = "none")
```

```
## Joining, by = c("fips", "name")
```

Median Household Income in America



Question 6

```
fips_inc %>%
  arrange(desc(estimate)) %>%
  slice(1:10)
```

```
## # A tibble: 10 x 5
##   fips      name      variable    estimate     moe
##   <chr>    <chr>    <chr>        <dbl>    <dbl>
## 1 51107 Loudoun County, Virginia median_hh_inc 142299  2089
## 2 51610 Falls Church city, Virginia median_hh_inc 127610 16144
## 3 51059 Fairfax County, Virginia median_hh_inc 124831  1281
## 4 06085 Santa Clara County, California median_hh_inc 124055  1117
## 5 06081 San Mateo County, California median_hh_inc 122641  1680
## 6 35028 Los Alamos County, New Mexico median_hh_inc 121324  4613
## 7 24027 Howard County, Maryland median_hh_inc 121160  2169
## 8 51013 Arlington County, Virginia median_hh_inc 120071  2064
## 9 08035 Douglas County, Colorado median_hh_inc 119730  1710
## 10 51600 Fairfax city, Virginia median_hh_inc 116979  7777
```

```
fips_inc %>%
  arrange(estimate) %>%
  slice(1:5)
```

```
## # A tibble: 5 x 5
##   fips      name      variable    estimate     moe
##   <chr>    <chr>    <chr>        <dbl>    <dbl>
## 1 72093 Maricao Municipio, Puerto Rico median_hh_inc 12441 3490
## 2 72055 Guánica Municipio, Puerto Rico median_hh_inc 12995 1031
## 3 72045 Comerío Municipio, Puerto Rico median_hh_inc 13733 1218
## 4 72079 Lajas Municipio, Puerto Rico median_hh_inc 13992 1071
## 5 72001 Adjuntas Municipio, Puerto Rico median_hh_inc 14012 1363
```