Chapter 2 Exercise

Meagan Fairfield-Peak

2/13/2022

Loading Libraries

```
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 tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

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

```
library(tidycensus)
# census_api_key("ea26feb1a3870fc890f9d802b8db001ebce20b7e", install = TRUE)
```

1. Review the available geographies in tidycensus from the geography table in this chapter. Acquire data on median age (variable B01002_001) for a geography we have not yet used.

Median Age of the Maryland County Subdivision.

```
median_age <- get_acs(
  geography = "county subdivision",
  variables = "B01002_001",
  state = "MD")
```

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

```
View(median_age)
```

2. Use the load_variables() function to find a variable that interests you that we haven't used yet. Use get_acs() to fetch data from the 2015-2019 ACS for counties in the state where you live, where you

have visited, or where you would like to visit.

PA Counties for variable B18102_029 which is Female: 18 to 34 years: With a hearing difficulty

```
v19 <- load_variables(2019, "acs5", cache = TRUE)
View(v19)

Pa_34yr_femalehearing <- get_acs(
  geography = "county",
  variables = "B18102_029",
  state = "PA")
```

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

```
View(Pa_34yr_femalehearing)
```

Chapter 3 Exercise

Meagan Fairfield-Peak

2/13/2022

Loading Libraries

```
library(tidycensus)
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 tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

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

The ACS Data Profile includes a number of pre-computed percentages which can reduce your data wrangling time. The variable in the 2015-2019 ACS for “percent of the population age 25 and up with a bachelor’s degree” is DP02_0068P. For a state of your choosing, use this variable to determine:

- The county with the highest percentage in the state
  - Chester County is the highest with 53% estimated
- The county with the lowest percentage in the state
  - Forest County is the lowest with 8.1% estimated
- The median value for counties in your chosen state
  - Median Value for PA Counties is 21.3%

```
bachelors_25up <- get_acs(
  geography = "county",
  state = "PA" ,
  variables = "DP02_0068P",
  year = 2019
)
```

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

```
## Using the ACS Data Profile
```

```
arrange(bachelors_25up, desc(estimate))
```

```
## # A tibble: 67 x 5
##    GEOID NAME                              variable   estimate   moe
##    <chr> <chr>                             <chr>         <dbl> <dbl>
##  1 42029 Chester County, Pennsylvania      DP02_0068P     53     0.6
##  2 42091 Montgomery County, Pennsylvania   DP02_0068P     49.3   0.4
##  3 42027 Centre County, Pennsylvania       DP02_0068P     45.5   1.1
##  4 42003 Allegheny County, Pennsylvania    DP02_0068P     41.6   0.3
##  5 42017 Bucks County, Pennsylvania        DP02_0068P     41.3   0.6
##  6 42045 Delaware County, Pennsylvania     DP02_0068P     39     0.6
##  7 42041 Cumberland County, Pennsylvania   DP02_0068P     36.6   0.9
##  8 42019 Butler County, Pennsylvania       DP02_0068P     36     0.8
##  9 42093 Montour County, Pennsylvania      DP02_0068P     31.6   2.6
## 10 42043 Dauphin County, Pennsylvania      DP02_0068P     30.8   1.1
## # ... with 57 more rows
```

```
arrange(bachelors_25up, estimate)
```

```
## # A tibble: 67 x 5
##    GEOID NAME                              variable   estimate   moe
##    <chr> <chr>                             <chr>         <dbl> <dbl>
##  1 42053 Forest County, Pennsylvania       DP02_0068P      8.1   1.9
##  2 42087 Mifflin County, Pennsylvania      DP02_0068P     11.9   1.2
##  3 42057 Fulton County, Pennsylvania       DP02_0068P     13.6   1.1
##  4 42023 Cameron County, Pennsylvania      DP02_0068P     13.7   2.6
##  5 42067 Juniata County, Pennsylvania      DP02_0068P     14.5   1.4
##  6 42009 Bedford County, Pennsylvania      DP02_0068P     14.7   1
##  7 42033 Clearfield County, Pennsylvania   DP02_0068P     15.1   0.8
##  8 42105 Potter County, Pennsylvania       DP02_0068P     15.4   1.1
##  9 42065 Jefferson County, Pennsylvania    DP02_0068P     15.9   1
## 10 42111 Somerset County, Pennsylvania     DP02_0068P     15.9   0.8
## # ... with 57 more rows
```

```
median(bachelors_25up$estimate)
```

```
## [1] 21.3
```

Chapter 4 Exercise

Meagan Fairfield-Peak

2/13/2022

Chapter 4 Exercise

Libraries

```
library(tidycensus)
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 tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

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

```
library(ggplot2)
```

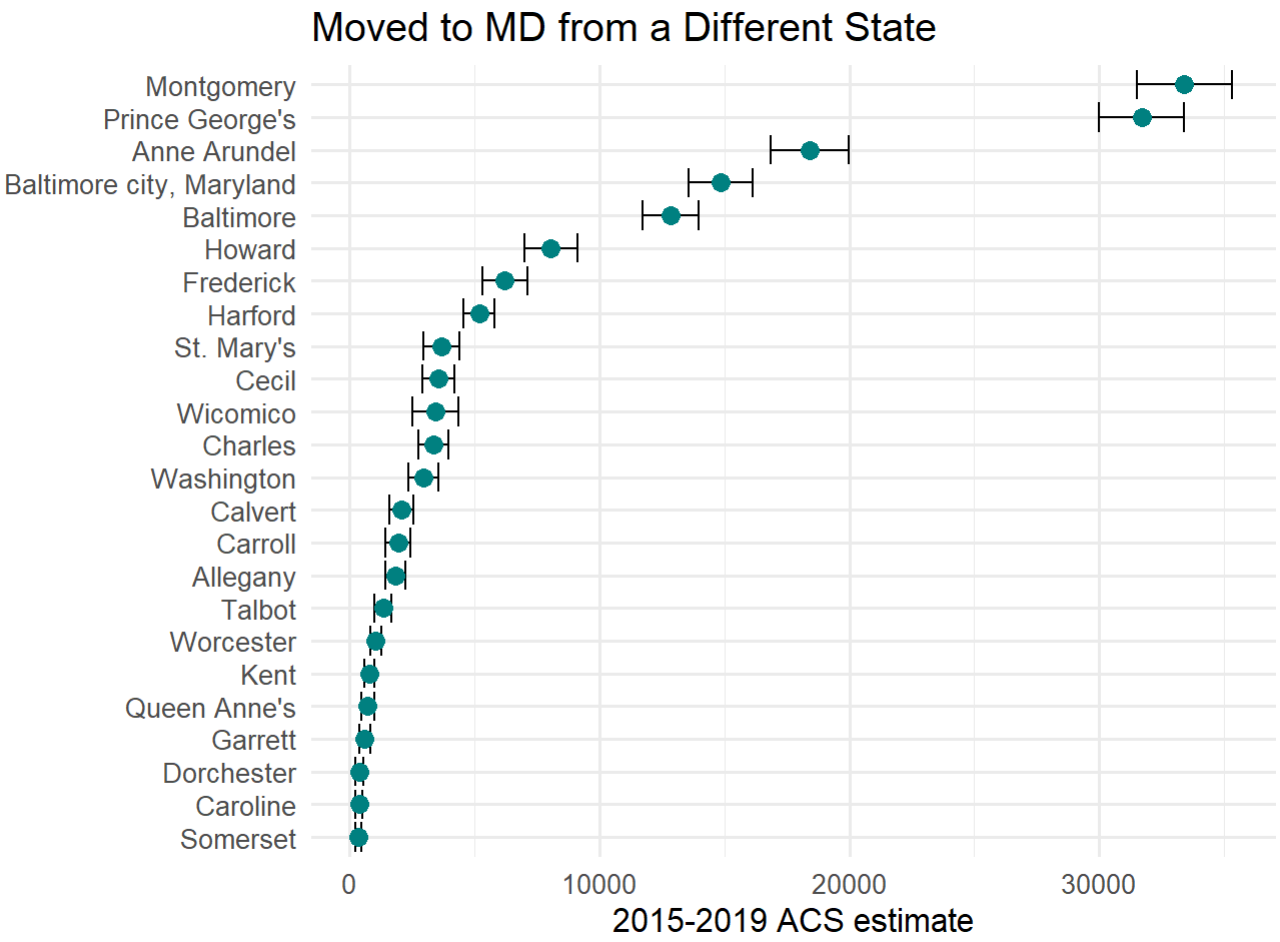
Choose a different variable in the ACS and/or a different location and create a margin of error visualization of your own.

```
v19 <- load_variables(2019, "acs5", cache = TRUE)
View(v19)

md_moving <- get_acs(
  state = "MD",
  geography = "county",
  variables = c(tot_diffstate = "B07003_013"),
  year = 2019
)%>%
  mutate(NAME = str_remove(NAME, " County, Maryland"))
```

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

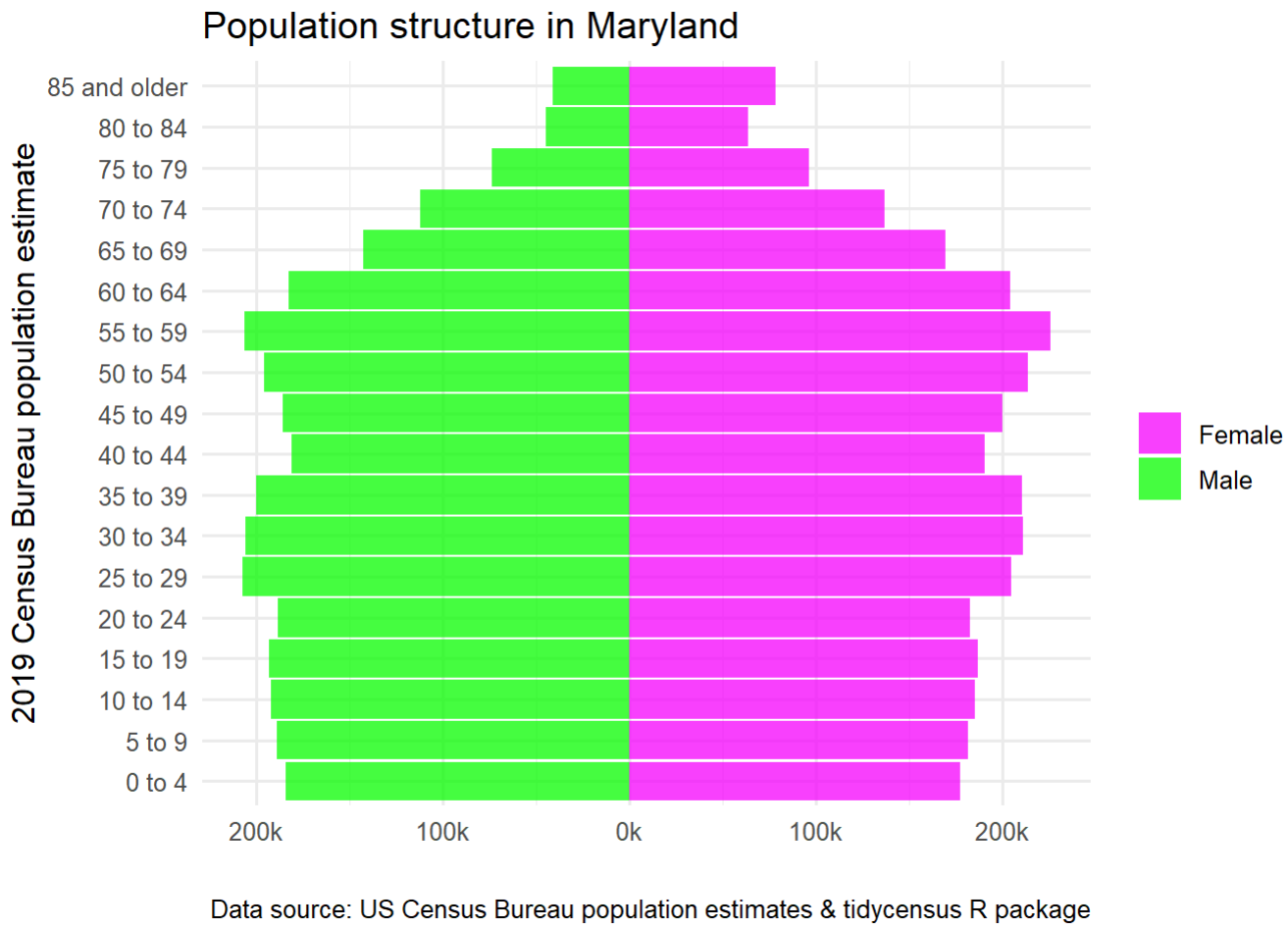
```
ggplot(md_moving, aes(x = estimate, y = reorder(NAME, estimate))) +
  geom_errorbarh(aes(xmin = estimate - moe, xmax = estimate + moe)) +
  geom_point(size = 3, color = "#008080") +
  theme_minimal(base_size = 12.5) +
  labs(title = "Moved to MD from a Different State",
       x = "2015-2019 ACS estimate",
       y = "")
```



Modify the population pyramid code to create a different, customized population pyramid. You can choose a different location (state or county), different colors/plot design, or some combination!

```
maryland <- get_estimates(
  geography = "state",
  state = "MD",
  product = "characteristics",
  breakdown = c("SEX", "AGEGROUP"),
  breakdown_labels = TRUE,
  year = 2019
)
md_filtered <- filter(maryland, str_detect(AGEGROUP, "^Age"),
                  SEX != "Both sexes") %>%
  mutate(value = ifelse(SEX == "Male", -value, value))
md_pyramid <- ggplot(md_filtered,
                       aes(x = value,
                           y = AGEGROUP,
                           fill = SEX)) +
  geom_col(width = 0.95, alpha = 0.75) +
  theme_minimal(base_family = "Verdana",
                base_size = 12) +
  scale_x_continuous(labels = function(x) paste0(abs(x / 1000), "k")) +
  scale_y_discrete(labels = function(y) str_remove_all(y, "Age\\s|\\syears")) +
  scale_fill_manual(values = c("#F500FC", "#07FC00")) +
  labs(x = "",
       y = "2019 Census Bureau population estimate",
       title = "Population structure in Maryland",
       fill = "",
       caption = "Data source: US Census Bureau population estimates & tidycensus R package")

md_pyramid
```


Population structure in Maryland
2019 Census Bureau population estimate
85 and older
80 to 84
75 to 79
70 to 74
65 to 69
60 to 64
55 to 59
50 to 54
45 to 49
40 to 44
35 to 39
30 to 34
25 to 29
20 to 24
15 to 19
10 to 14
5 to 9
0 to 4
200k
100k
0k
100k
200k
Female
Male
Data source: US Census Bureau population estimates & tidycensus R package