Project 3: Demographics

TOTAL: 20pts

PURPOSE

To understand patterns of demographics across the U.S. and use those patterns to understand similarities and differences to Washington, D.C.

GOAL

Analyze census data across the U.S. to identify areas (cities/counties/metropolitan statistical areas) similar to Washington, D.C.

DATA

The U.S. Census Bureau's American Community Survey Data.

METHODS

Shout-out to this blog for making sense of the tidycensus package.

Step 0: Libraries

1. You will need to install these packages (run both of these if using the AU computers):
    a. install.packages('tidyverse')
    b. install.packages('tidycensus')
2. You will need to load these libraries, no matter the computer:
    a. library(tidyverse)
    b. library(tidycensus)

Step 1: Census API key

1. First, get an API key:
    a. Go here
    b. Sign-up. Check your email. Get your API key.
2. Second, install and load this package and your API key:

a. census_api_key("YOUR API KEY GOES HERE", install = TRUE, overwrite = TRUE) # run once!
   i. If you experience API discomfort, use this:

b. **Now you must run this line!**
   i. readRenviron("~/.Renviron")

Step 2: Find variables

1. Create a table of variables to use
   a. # all the variables
   b. variables <- load_variables(2023, "acs5", cache = TRUE)
   c. # just the totals
   d. variables.totals <- subset(variables, variables$label == 'Estimate!!Total:')
2. Open the table, turn on the filter, and search for variables of interest
   a. Spend some time here identifying **10+ variables of interest**

Step 3: Write queries

Some examples:

1. Run a query for 2023 total population by state, like this:
   a. pop23 <- get_acs(geography = "state",
   b. variables = "B01003_001",
   c. year = 2023)
2. Run a query for 2023 total population by metropolitan statistical area, like this:
   a. pop.msa.23 <- get_acs(geography = "metropolitan statistical area/micropolitan statistical area",
   b. variables = "B01003_001",
   c. year = 2023)
   d. # compare to DC
   e. pop.msa.23$diff <- abs(pop.msa.23$estimate - pop.msa.23$estimate[xxx])
   f. # to make this table easier to work with
   g. population <- subset(pop.msa.23, pop.msa.23$diff < xxxxxx)
3. Run a query for 2023 population by race by MSA, like this:
   a. msa.race <- get_acs(geography = "metropolitan statistical area/micropolitan statistical area",
   b. variables = c("B02008_001", "B02009_001", "B02010_001", "B02011_001"),
   c. year = 2023)
   d. # replace with codes with text
   e. msa.race$variable <- gsub("B02008_001", "White", msa.race$variable)
   f. msa.race$variable <- gsub("B02009_001", "Black", msa.race$variable)
   g. msa.race$variable <- gsub("B02010_001", "American Indian", msa.race$variable)
   h. msa.race$variable <- gsub("B02011_001", "Asian", msa.race$variable)
   i. # join population and race tables
   j. msa.race.total <- msa.race %>%
   k. left_join(pop.msa.23, by = "GEOID")

```
# clean up the columns
msa.race.total <- msa.race.total[c(1:4,8,10)]
# clean up the column names
names(msa.race.total) <- c("GEOID", "Name", "Race", "Race.Count",
                           "Population", "Pop.Diff")
# calculate race percentages
msa.race.total$race.PCT <-
round(msa.race.total$Race.Count/msa.race.total$Population*100,2)
# to create the new difference columns for comparison to DC...
# separate the tables
msa.race.black <- subset(msa.race.total, msa.race.total$Race == 'Black')
msa.race.white <- subset(msa.race.total, msa.race.total$Race == 'White')
# calculate differences
msa.race.black$race.diff <- abs(msa.race.black$race.PCT -
msa.race.black$race.PCT[xxx])
msa.race.white$race.diff <- abs(msa.race.white$race.PCT -
msa.race.white$race.PCT[xxx])
# reconstruct the table
msa.race.all <- rbind(msa.race.black, msa.race.white)
```

4. Run a query for 2023 'component' datasets by state, like this:
   a. "The variables included in the components of change product consist of both estimates of counts and rates. Rates are preceded by an R in the variable name and are calculated per 1000 residents."

```
state_components <- get_estimates(geography = "state", product = "components")
```

5. Run a query for 2023 'component' datasets by MSA, like this:

```
msa_components <- get_estimates(geography = "metropolitan statistical
area/micropolitan statistical area", product = "components")
```

6. Run a query for 2023 county comparisons of sex, age, and hispanic populations, like this:

```
sex.age.hisp <- get_estimates(geography = "county",
                              product = "characteristics",
                              breakdown = c("SEX", "AGEGROUP", "HISP"),
                              breakdown_labels = TRUE)
```

7. Run a query for 2023 county comparisons of sex, like this:

```
sex.county <- get_estimates(geography = "county",
                            product = "characteristics",
                            breakdown = c("SEX"),
                            breakdown_labels = TRUE)
```

8. For age, as an example:

```
county.age <- get_estimates(geography = "county",
                            product = "characteristics",
                            breakdown = c("AGEGROUP"),
                            breakdown_labels = TRUE)
```

9. Run a query for 2023 migration population flows to and from an MSA, like this:

```
dc_flows <- get_flows(
  geography = "metropolitan statistical area",
  msa = 47900, # can adjust this to different areas
  year = 2020,
  geometry = TRUE)
```

HERE IS THE SCRIPT WE BUILT IN CLASS ON 26 FEBRUARY 2025

- For details on the different census variables to analyze, this is a great resource
    - In SEARCH bar, put the variable code or keyword(s) you're interested in
        - Some datasets to consider:
            - Total population
            - Population count/percentage by Race
            - Migration flows
            - Income (B19019_001) = Median Household Income
            - Education (C15010_001) = number of Bachelor's degree
            - Poverty (B17026_001) = number of families at/below poverty

Step 4: Visualize data

- Use the explore package
- Build bar graphs like you did in Project 1.
- Build bar graphs and maps like you did in Project 2.
- Create scatter plots as necessary from the tables generated.
    - As an example:

```
# scatter plot
ggplot(age.similar) +
  geom_point(aes(x = value, y = NAME))
# bar chart
ggplot(population, aes(x= NAME, y= estimate)) +
  coord_flip() +
  geom_bar(stat="identity")
```

- **Do some analysis using the scatter plots to compare D.C. to other MSAs and counties**

HERE IS THE SCRIPT WE BUILT IN CLASS ON 05 MARCH 2025

Step 5: Develop a workflow/process

It's highly recommended to use a clear, repeatable process for finding your similar cities. Something, potentially, like this:

1. **Find data**
    a. Use the variables table to discover datasets to work with
2. **Get data**
    a. Write queries to generate new tables
3. **Finesse data**
    a. Replace codes with plain English descriptions using gsub
    b. Compare areas to DC using a diff column (remember there are 939 MSAs, and DC is usually listed last)
    c. Subset your data based on a specific field/attribute
        i. For example, if you queried population by race, filter your data to only look at a specific race
    d. Subset again to look at fewer areas
        i. This subset is based on the values within a field

ii. This creates a table with less rows that is likely great for visualizing

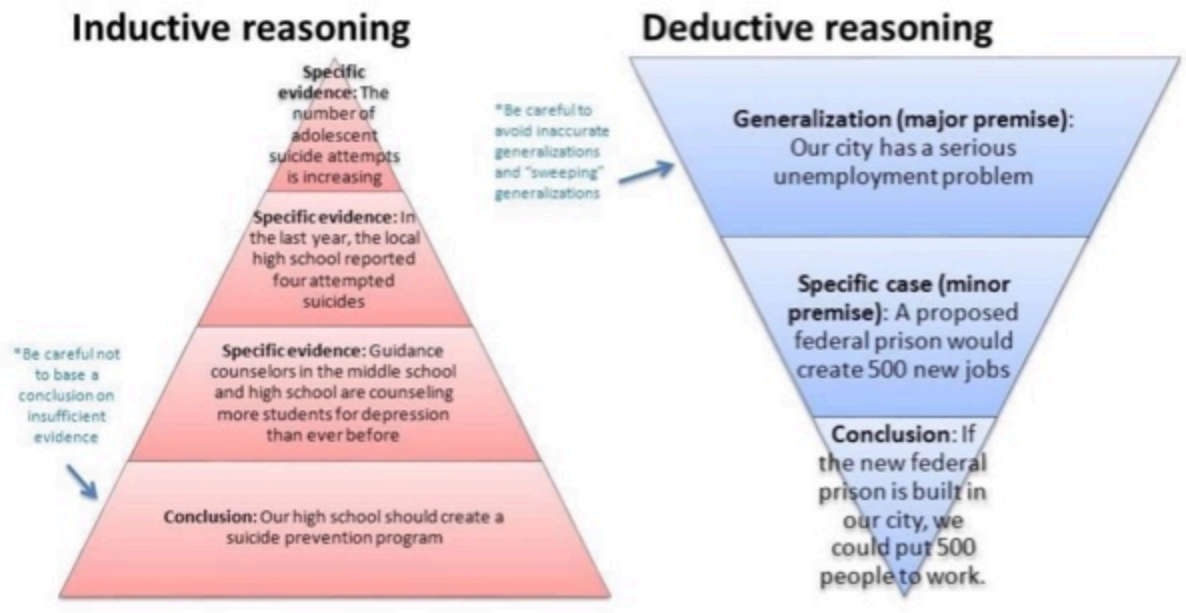
4. **Visualize data**

ANALYSIS

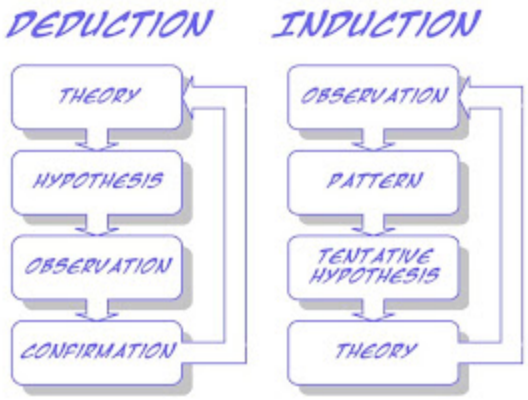
When conducting your analysis, consider this perspective:

1. State a finding
    a. This is the new thing that you've identified from the data
2. Provide your evidence
    a. This is typically a statistical reference
3. Add the context
    a. This is the integration of the stat(s) and the finding
    b. Compare your finding to the norm/average, or a similar measure in the data

When analyzing data, there's really two approaches: **Inductive** and **Deductive**



- **Inductive** is like a cone, where you start your process with a known or suspected value, and work outwards thru the data to identify evidence that supports it
    - Inductive is high-risk, high-reward. You will potentially find your answers either much faster, or much slower
- **Deductive** is like a funnel, where you start your process with everything, and filter thru the data to identify evidence so a specific case(s) naturally emerge
    - Deductive is even-paced. You will find your answers consistently and systematically.



Deductively, for this project, one way to conduct your analysis would be this:

1. Get your six datasets
2. Identify the top ***x*** cities most similar to D.C. in each
    a. ***x*** may be 5, 8, 10 top cities - your call
3. Compare the lists and find cities that consistently show up on different lists

Inductively, for this project, would be starting with a suspected similar city, identifying data likely to be related, and analyzing from there.

Your goal, for this project:

1. **Five similar "cities"**
    a. Consider cities, counties, and MSAs
        i. If you use counties or MSAs, be clear to address what major city is within that county
2. **Two "datasets" per similar city**
    a. Don't use two attributes or fields from the same dataset for one city (i.e. race or sex). Use completely different datasets.
    b. This is ten total datasets, unless you use a different attribute from the same dataset for different cities
3. **Analyze locations**
    a. Make maps
    b. Characterize any potential similar city as either being in the same hotspot as D.C., a hotspot in general, or not in a hotspot
        i. If a potential similar city is not located in a police shooting hotspot, it's not similar to D.C.
4. **One visual per city**
    a. You don't *need* to include maps as your visuals

FORMAT

Submit your script as a usable HTML file, via RMarkdown. You create this output as a .Rmd file in RStudio, but only submit the .html file.

Use this file as a template.

SUBMISSION

Once your analysis is complete, please submit your project as either an HTML file (as the output/export/knit of your RMarkdown script), via Canvas.

GRADES

Data: **2pt**

- Provide a brief paragraph (3-4 sentences) on the data used in this project. This includes the specific source(s), the size, and the specific temporal and spatial constraints.

Methods: **2pt**

- Provide a brief paragraph (3-4 sentences) on the research methods leveraged to answer this question. This includes the software, calculations, skills, techniques, and unique workflows used to analyze your data and develop an answer. You do NOT need to describe click-by-click instructions or lines of code describing how you did things; you DO need to describe a logical process that is specific enough that a reader could replicate.

Analysis: **10pts**

- Using the data specific to this project, identify five cities *similar* to Washington, D.C. (2pts each)
    - Provide justification of similarity for each city (4-5 sentences each)
        - Each justification should include at least two specific reasons for similarity
            - Each reason should include a relevant statistical reference

Visuals: **5pts**

- Create one visual per similar city (1pt each)
- Each visual should have a direct connection to the analysis, and be clearly labeled

Formatting: **1pt**

- Error-free writing. No typos, run-on sentences, or sentence fragments. Proper punctuation. Real words.
    - Need help paraphrasing dense content? Try this. And here's a great reference, too.
- Project created in RMarkdown and submitted as an HTML file via Canvas.

Please email me with any questions.