Hispanic Origins

ML

1/13/2020

Getting Started

Start by loading the packages. They are already loaded from our previous notebook, but to knit the file properly they have to be reloaded in each notebook.

```
library(tidyverse)
library(tidycensus)
```

We will use the `get_acs()` function to get data from the 2018 1-year ACS. But we first need to figure out what variables are available. We can do that by creating an object called `vars_acs18` and the `load_variables()` function.

```
vars_acs18 <- load_variables(year = 2018,
                             dataset = "acs1", #use dataset for load_variables
                             cache = TRUE)
```

Open the `vars_acs18` data frame and search for "Hispanic or Latino Origin". What table do we want to get the specific origins?

We could ask for specific variables (using the `variables =` command) or the full table. In this example, it is more effecient to get the full table.

```
hispanic_df <- get_acs(geography = "us",
                       table = "B03001",
                       year = 2018,
                       survey = "acs1") # use survey for get_acs
```

```
## Getting data from the 2018 1-year ACS
```

```
## The one-year ACS provides data for geographies with populations of 65,000 and greater.
```

Remember to change the variable names to lower case!

```
names(hispanic_df) <- tolower(names(hispanic_df))
```

We don't want all the variables. Specifically, we want to filter out the first three total variables (the overall total, the total Not Hispanics, and the the total Hispanics), as well as the subtotals for Central American (B03001_008), South American (B03001_017), and Other Hispanic or Latino (B03001_027).

```
hispanic_df <- hispanic_df %>%
  filter(variable != "B03001_001" & variable != "B03001_002" &
           variable != "B03001_003" &
           variable != "B03001_008" & variable != "B03001_016" &
           variable != "B03001_027")
```

Next we'll replace the variable names with labels.

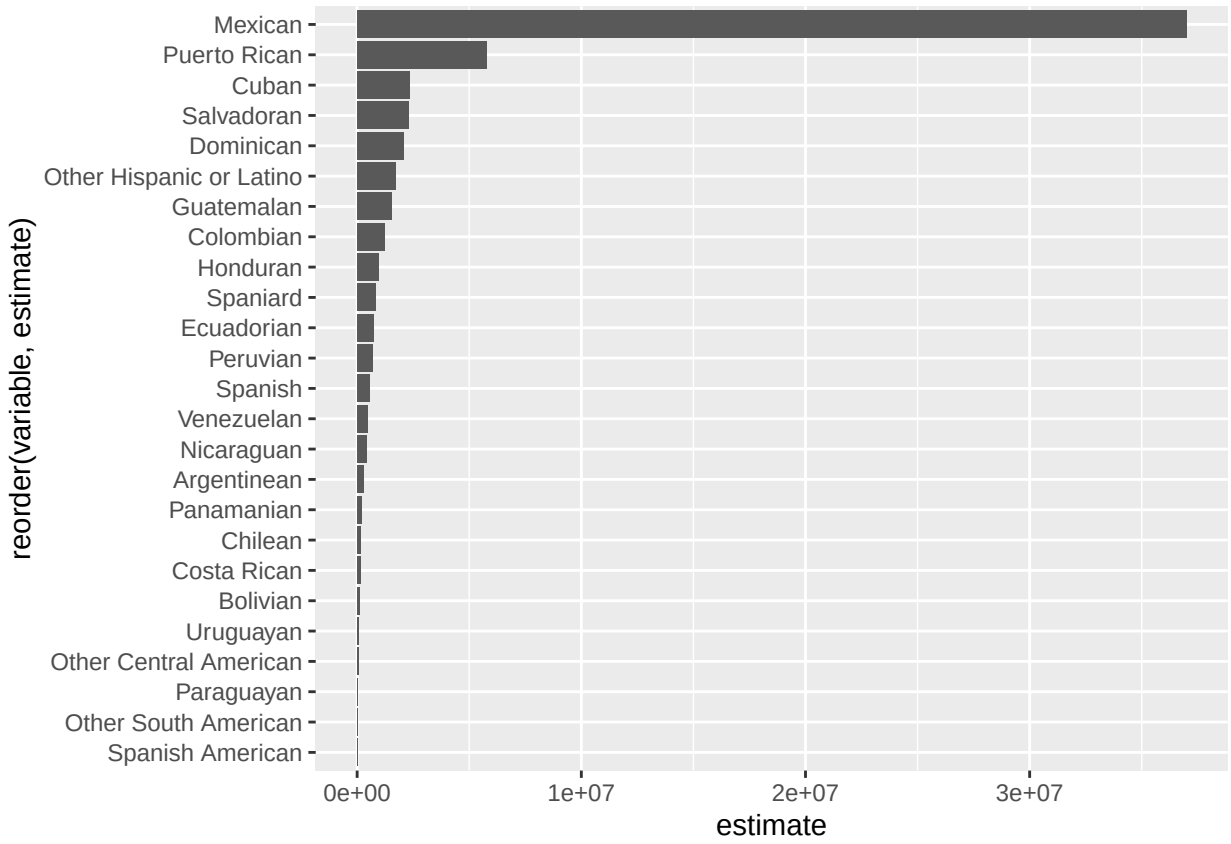
```
hispanic_df <- hispanic_df %>%
  mutate(variable = factor(variable,
                           labels = c("Mexican",
                                      "Puerto Rican",
                                      "Cuban",
```

```
                                    "Dominican",
                                    "Costa Rican",
                                    "Guatemalan",
                                    "Honduran",
                                    "Nicaraguan",
                                    "Panamanian",
                                    "Salvadoran",
                                    "Other Central American",
                                    "Argentinean",
                                    "Bolivian",
                                    "Chilean",
                                    "Colombian",
                                    "Ecuadorian",
                                    "Paraguayan",
                                    "Peruvian",
                                    "Uruguayan",
                                    "Venezuelan",
                                    "Other South American",
                                    "Spaniard",
                                    "Spanish",
                                    "Spanish American",
                                    "Other Hispanic or Latino")))
```

Create a plot showing the number of people with each Hispanic origin. Order the origins by number of people.

REPLACE THIS LINE WITH YOUR CODE

```
hispanic_origins_plot <- ggplot(hispanic_df,
                                aes(x = reorder(variable, estimate),
                                    y = estimate))

hispanic_origins_plot + geom_col() + coord_flip()
```

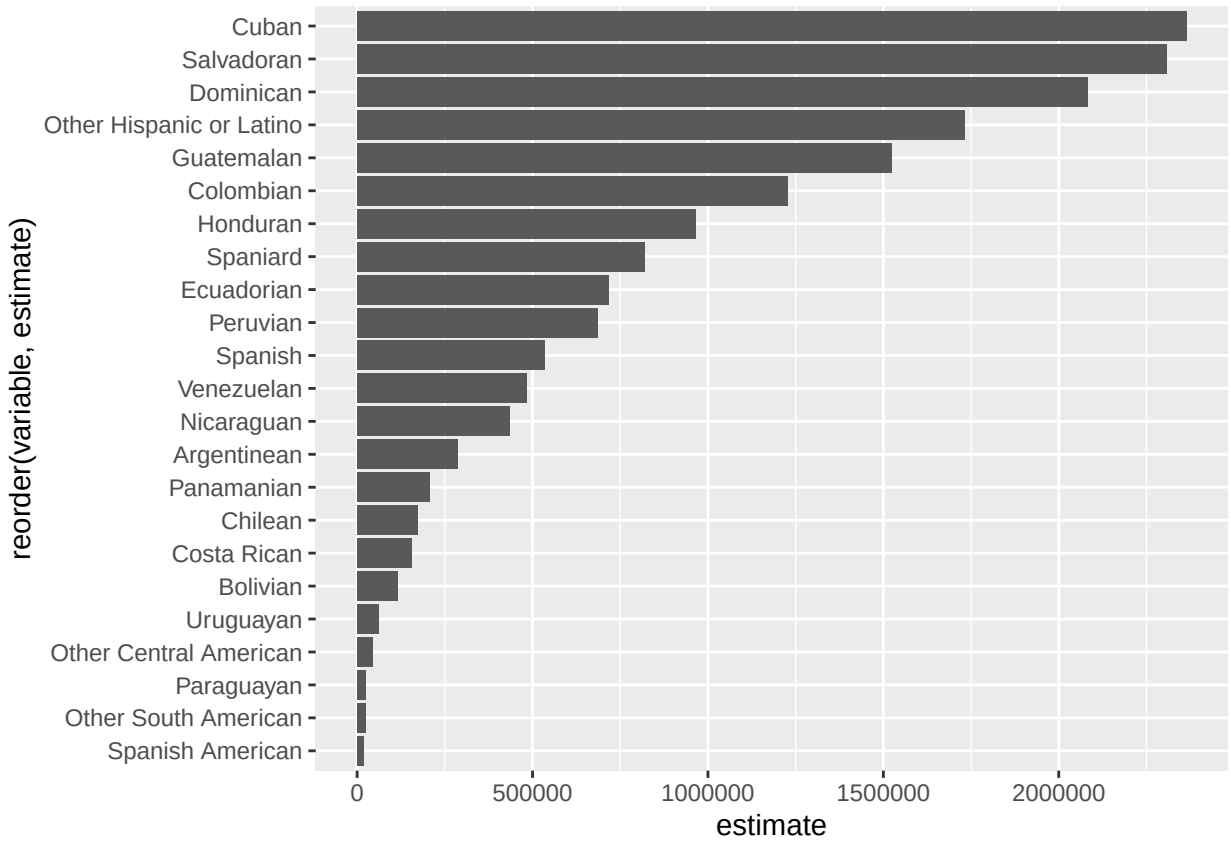


It is pretty difficult here to see the numbers for the groups that have smaller populations than Mexicans and Puerto Ricans. Filter out those two groups from a new data frame called `hispanic_nomexpr_df` and redo the plot.

REPLACE THIS LINE WITH YOUR CODE

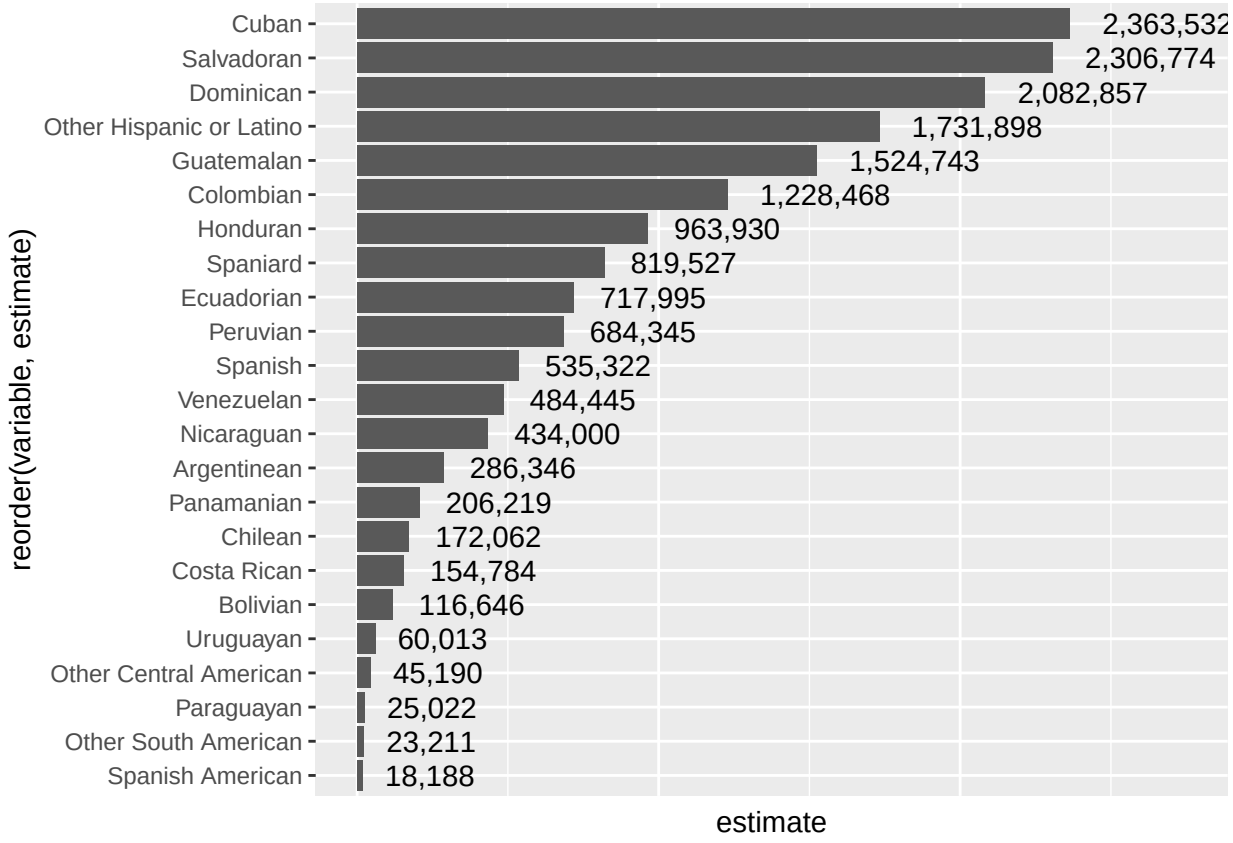
```
hispanic_nomexpr_df <- hispanic_df %>%
  filter(variable!="Mexican" & variable!="Puerto Rican")
```

```
hispanic_nomexpr_plot <- ggplot(hispanic_nomexpr_df,
                                aes(x = reorder(variable, estimate),
                                    y = estimate))

hispanic_nomexpr_plot + geom_col() + coord_flip()
```



Some Improvements

```
hispanic_nomexpr_plot <- ggplot(hispanic_nomexpr_df,
                                aes(x = reorder(variable, estimate),
                                    y = estimate,
                                    label = scales::comma(estimate)))

hispanic_nomexpr_plot + geom_col() + coord_flip() +
  geom_text(hjust = -.25) +
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank()) +
  ylim(c(0,2750000))
```



Introducing Mapping

The mapping functions built into ggplot require geometric shape files. Fortunately, they are easy to get using `tidycensus`. We can download them by adding `geometry = TRUE` to our `get_acs()` (or `get_estimates` or `get_decennial`) function. The `shift_geo = TRUE` moves Alaska and Hawaii so they are easier to see.

In this example, we want data for each county. We could also use state or tracts or blocks, and we can get those for specific states. Which survey should we use for county-level data?

```
hispanic_geo <- get_acs(geography = "county",
                        #state = "Vermont", # if we only want VT counties
                        table = "B03001",
                        year = 2018,
                        survey = "acs5",
                        geometry = TRUE,
                        shift_geo = TRUE)
```

```
## Getting data from the 2014-2018 5-year ACS
```

```
## Using feature geometry obtained from the albersusa package
```

```
## Please note: Alaska and Hawaii are being shifted and are not to scale.
```

Repeat all the cleanup we did when using the national data.

```
names(hispanic_geo) <- tolower(names(hispanic_geo))

hispanic_geo <- hispanic_geo %>%
  filter(variable != "B03001_001" & variable != "B03001_002" &
```

```
          variable != "B03001_003" &
          variable != "B03001_008" & variable != "B03001_016" &
          variable != "B03001_027") %>%
  mutate(variable = factor(variable,
                           labels = c("Mexican",
                                      "Puerto Rican",
                                      "Cuban",
                                      "Dominican",
                                      "Costa Rican",
                                      "Guatemalan",
                                      "Honduran",
                                      "Nicaraguan",
                                      "Panamanian",
                                      "Salvadoran",
                                      "Other Central American",
                                      "Argentinean",
                                      "Bolivian",
                                      "Chilean",
                                      "Colombian",
                                      "Ecuadorian",
                                      "Paraguayan",
                                      "Peruvian",
                                      "Uruguayan",
                                      "Venezuelan",
                                      "Other South American",
                                      "Spaniard",
                                      "Spanish",
                                      "Spanish American",
                                      "Other Hispanic or Latino"))) %>%
  filter(variable!="Mexican" & variable!="Puerto Rican")
```

Our data frame has the esimated number of people with each origin in each county. We want a new data frame called `hispanic_geo_summary` with the origin with the highest number of people in each county. How do we do this?

REPLACE THIS LINE WITH YOUR CODE

```
hispanic_geo_summary <- hispanic_geo %>%
  group_by(name) %>%
  filter(estimate == max(estimate))
```

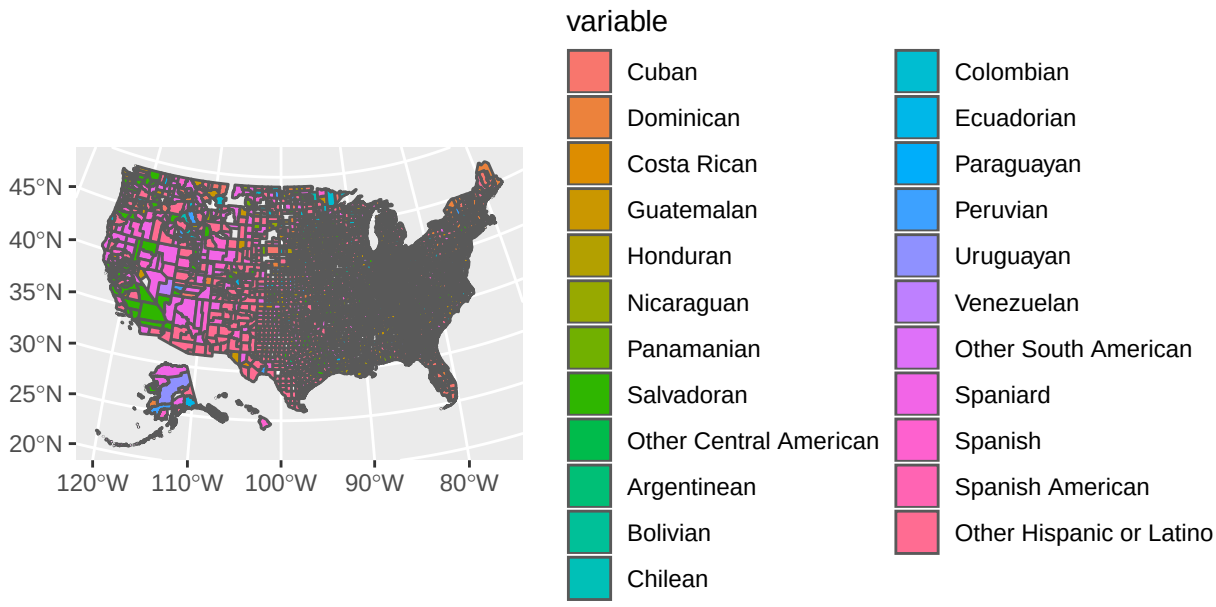
Our new data frame has more observations than there are counties. Why? How do we fix this?

REPLACE THIS LINE WITH YOUR CODE

```
hispanic_geo_summary <- hispanic_geo_summary %>%
  filter(estimate != 0)
```

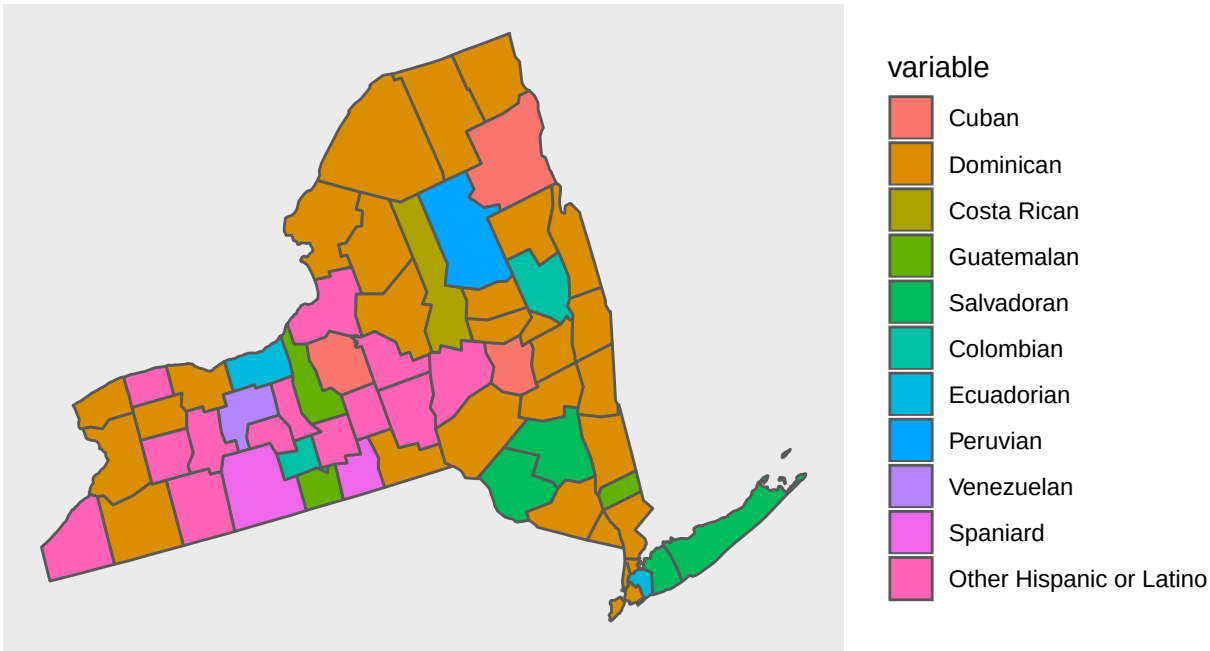
We are ready to make our map. Set it up like a regular ggplot figure by giving the data frame name. We don't use x and y variables here but will want an aesthetic map that says what variable we want to use to fill in the counties. The `geom_sf()` function adds the "simple feature" of a map.

```
hispanic_map <- ggplot(hispanic_geo_summary, aes(fill = variable))

hispanic_map + geom_sf()
```



There's a lot going on here. Let's just look at counties in the state of New York.

```
one_state <- hispanic_geo_summary %>%
  filter(str_detect(name, ", New York"))

ggplot(one_state, aes(fill = variable)) + geom_sf() + coord_sf(datum = NA)
```



Extra

```
library(mapview)
mapview(one_state, zcol = "variable", legend = FALSE)
```