Tidycensus Function

2023-06-06

## Creating a Function to Automatically Format Race and Ethnicity Data

Let’s take a look at one more custom function that I’ve created. As we saw in Chapter @ref(tidycensus-chapter), we can use the tidycensus package to easily pull in data directly from the United States Census Bureau. As we also saw in that same chapter, it can be hard to remember the names of variables. I regularly want to access data on population by race and ethnicity from the American Community Survey, but I never remember which variables I need to access to do so. I’ve created the get\_acs\_race\_ethnicity() function to help me.

library(tidycensus)  
  
get\_acs\_race\_ethnicity <- function(...) {  
   
 get\_acs(...,  
 variables = c("White" = "B03002\_003",  
 "Black/African American" = "B03002\_004",  
 "American Indian/Alaska Native" = "B03002\_005",  
 "Asian" = "B03002\_006",  
 "Native Hawaiian/Pacific Islander" = "B03002\_007",  
 "Other race" = "B03002\_008",  
 "Multi-Race" = "B03002\_009",  
 "Hispanic/Latino" = "B03002\_012"))   
   
}

Much of the code in this function will look familiar: it uses the get\_acs() function to bring in data. And it gives names to the variables (for example, “Asian” instead of “B03002\_006”). But the novel piece is the ... (pronounced “dot-dot-dot”) used within function() and within get\_acs(). To explain what this is, let’s talk about why it exists. I created the get\_acs\_race\_ethnicity() function to do two things:

1. Bring in data on population by race and ethnicity.
2. Give my variables more meaningful names.

What get\_acs\_race\_ethnicity() does not do is determine the geography needed. Do I want to get data at the state level, say, the county level, or the Census tract level? I could add an argument for this, but I’d just be recreating arguments that already exist in the get\_acs() function. See below for what this would look like.

get\_acs\_race\_ethnicity <- function(my\_geography) {  
   
 get\_acs(geography = my\_geography,  
 variables = c("White" = "B03002\_003",  
 "Black/African American" = "B03002\_004",  
 "American Indian/Alaska Native" = "B03002\_005",  
 "Asian" = "B03002\_006",  
 "Native Hawaiian/Pacific Islander" = "B03002\_007",  
 "Other race" = "B03002\_008",  
 "Multi-Race" = "B03002\_009",  
 "Hispanic/Latino" = "B03002\_012"))   
   
}

As we saw in Chapter @ref(tidycensus-chapter), the get\_acs() function has many arguments. I don’t want to repeat them all as arguments in my get\_acs\_race\_ethnicity() function, and the ... allows us to avoid doing so. By putting ... in the get\_acs\_race\_ethnicity() function, any arguments listed there are then passed to the get\_acs() function (note the ... there as well). I can then run this code to set the geography to state.

get\_acs\_race\_ethnicity(geography = "state")

Which would return the following:

#> # A tibble: 416 × 5  
#> GEOID NAME variable estimate moe  
#> <chr> <chr> <chr> <dbl> <dbl>  
#> 1 01 Alabama White 3241003 2076  
#> 2 01 Alabama Black/African American 1316314 3018  
#> 3 01 Alabama American Indian/Alaska Nati… 17417 941  
#> 4 01 Alabama Asian 69331 1559  
#> 5 01 Alabama Native Hawaiian/Pacific Isl… 1594 376  
#> 6 01 Alabama Other race 12504 1867  
#> 7 01 Alabama Multi-Race 114853 3835  
#> 8 01 Alabama Hispanic/Latino 224659 413  
#> 9 02 Alaska White 434515 1067  
#> 10 02 Alaska Black/African American 22787 769  
#> # ℹ 406 more rows

I could run my get\_acs\_race\_ethnicity() function again and add geometry = TRUE as follows.

get\_acs\_race\_ethnicity(geography = "state",  
 geometry = TRUE)

My get\_acs\_race\_ethnicity() function will now return geospatial data alongside my demographic data.

#> Simple feature collection with 416 features and 5 fields  
#> Geometry type: MULTIPOLYGON  
#> Dimension: XY  
#> Bounding box: xmin: -179.1489 ymin: 17.88328 xmax: 179.7785 ymax: 71.36516  
#> Geodetic CRS: NAD83  
#> First 10 features:  
#> GEOID NAME variable estimate  
#> 1 56 Wyoming White 478508  
#> 2 56 Wyoming Black/African American 4811  
#> 3 56 Wyoming American Indian/Alaska Native 11330  
#> 4 56 Wyoming Asian 4907  
#> 5 56 Wyoming Native Hawaiian/Pacific Islander 397  
#> 6 56 Wyoming Other race 1582  
#> 7 56 Wyoming Multi-Race 15921  
#> 8 56 Wyoming Hispanic/Latino 59185  
#> 9 02 Alaska White 434515  
#> 10 02 Alaska Black/African American 22787  
#> moe geometry  
#> 1 959 MULTIPOLYGON (((-111.0546 4...  
#> 2 544 MULTIPOLYGON (((-111.0546 4...  
#> 3 458 MULTIPOLYGON (((-111.0546 4...  
#> 4 409 MULTIPOLYGON (((-111.0546 4...  
#> 5 158 MULTIPOLYGON (((-111.0546 4...  
#> 6 545 MULTIPOLYGON (((-111.0546 4...  
#> 7 1098 MULTIPOLYGON (((-111.0546 4...  
#> 8 167 MULTIPOLYGON (((-111.0546 4...  
#> 9 1067 MULTIPOLYGON (((179.4825 51...  
#> 10 769 MULTIPOLYGON (((179.4825 51...

The ... allows me to create my own function and pass any arguments from it to another function without repeating all of the child function’s arguments, giving me flexibility while maintaining brevity.