tidycensus

tidycensus is an R package that allows users to interface with a select number of the US Census Bureau's data APIs and return tidyverse-ready data frames, optionally with simple feature geometry included.

To get started working with tidycensus, users should load the package along with the tidyverse package, and set their Census API key. A key can be obtained from http://api.census.gov/data/key_signup.html.

Loading tidycensus package

```
library(tidycensus)
library(tidyverse)
library(insight) #package to create exportable tables

#census_api_key("YOUR KEY GOES HERE")
```

If you need to install the tidycensus package, run this line in the console: ***install.packages("tidycensus")***

Using the tidycensus package

The tidycensus package allows users to call on both decennial US Census APIs (2000, 2010, and 2020) and the 1-year and 5-year American Community Survey APIs.

- get_decennial()
- get_acs()

Using `get_decennial()` First, we will look at an example of using the decennial census. We will pull median age for counties in Connecticut in 2020 from the Demographic and Housing Characteristics summary file. If you wanted to pull data by towns in Connecticut you would use `geography = "county subdivision"`.

To search variables you can use the `load_variables()` function. For the decennial Census, possible dataset choices include `"pl"` for the redistricting files; `"dhc"` for the Demographic and Housing Characteristics file and `"dp"` for the Demographic Profile (2020 only).

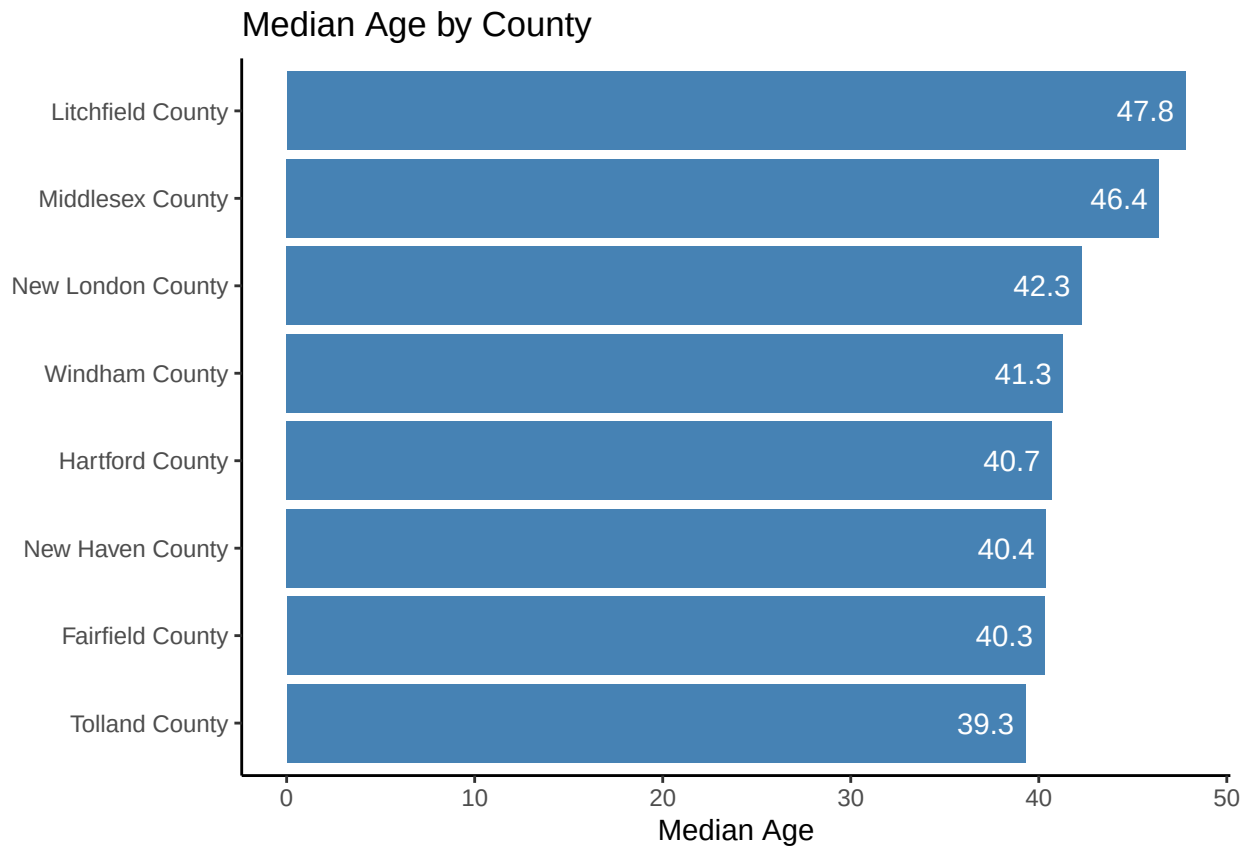
```
dec20_vars <- load_variables(2020, "dhc", cache = TRUE)
```

```
median_age_dec <- get_decennial(
  geography = "County",
  #geography = "county subdivision",
  state = "Connecticut",
  variables = "P13_001N",
  year = 2020,
  sumfile = "dhc")
```

Here is a simple bar chart of the median age by county made with the ggplot2 package.

```r
#Dropping Connecticut from the name for a cleaner axis label on chart
median_age_dec_v2 <- median_age_dec %>%
  separate(NAME, sep = ",", c("County", NA))

median_age_dec_v2 %>%
  ggplot(aes(x = value, y = reorder(County, value))) +
  geom_bar(stat="identity", fill="steelblue") +
  labs(
    title = "Median Age by County",
    x = "Median Age",
    y = NULL
    ) +
  geom_text(aes(label = value), hjust = 1.2, colour = "white")+
  theme_classic()
```



Next, this is an example of pulling a whole table rather than specific variables. We will still look at Median Age but now break it down by sex.

```r
median_age_dec_table <- get_decennial(
  geography = "County",
  state = "Connecticut",
  table = "P13",
  year = 2020,
  sumfile = "dhc")
```

Instead of pulling the table in a long-format (the default), you can use `output = "wide"` to format the data

so each variable has it's own column.

```
median_age_dec_table <- get_decennial(
  geography = "County",
  state = "Connecticut",
  table = "P13",
  year = 2020,
  sumfile = "dhc",
  output = "wide")
```

Now let's create a output table for Median Age by Sex. We will use the insight package to create an exportable formatted table.

```
#Dropping Connecticut from the name for a cleaner chart
median_age_dec_table_v2 <- median_age_dec_table %>%
  separate(NAME, sep = ",", c("County", NA)) %>%
  rename(`Median Age All` = P13_001N, `Median Age Male`  = P13_002N, `Median Age Female`  = P13_003N) %
  select(!GEOID)

export_table(median_age_dec_table_v2, format = "md") #using the Insights package and markdown format
```

County	Median Age All	Median Age Male	Median Age Female
Fairfield County	40.30	38.90	41.60
Hartford County	40.70	39.10	42.30
Litchfield County	47.80	46.40	49.00
Middlesex County	46.40	44.90	47.80
New Haven County	40.40	38.90	41.80
New London County	42.30	40.50	44.10
Tolland County	39.30	38.10	40.50
Windham County	41.30	40.20	42.70

Using `get_acs()` Next, we will look at an example of using the American Community Survey (ACS). We will pull median household income for the COGs (county equivalents) in Connecticut in 2022 from the 5-Year ACS.

To search variables you can use the `load_variables()` function. For the 1-year estimates,include `"acs1"`. For the 5-year estimates, include`"acs5"`.

```
ACS22_vars <- load_variables(2022, "acs5", cache = TRUE)
```

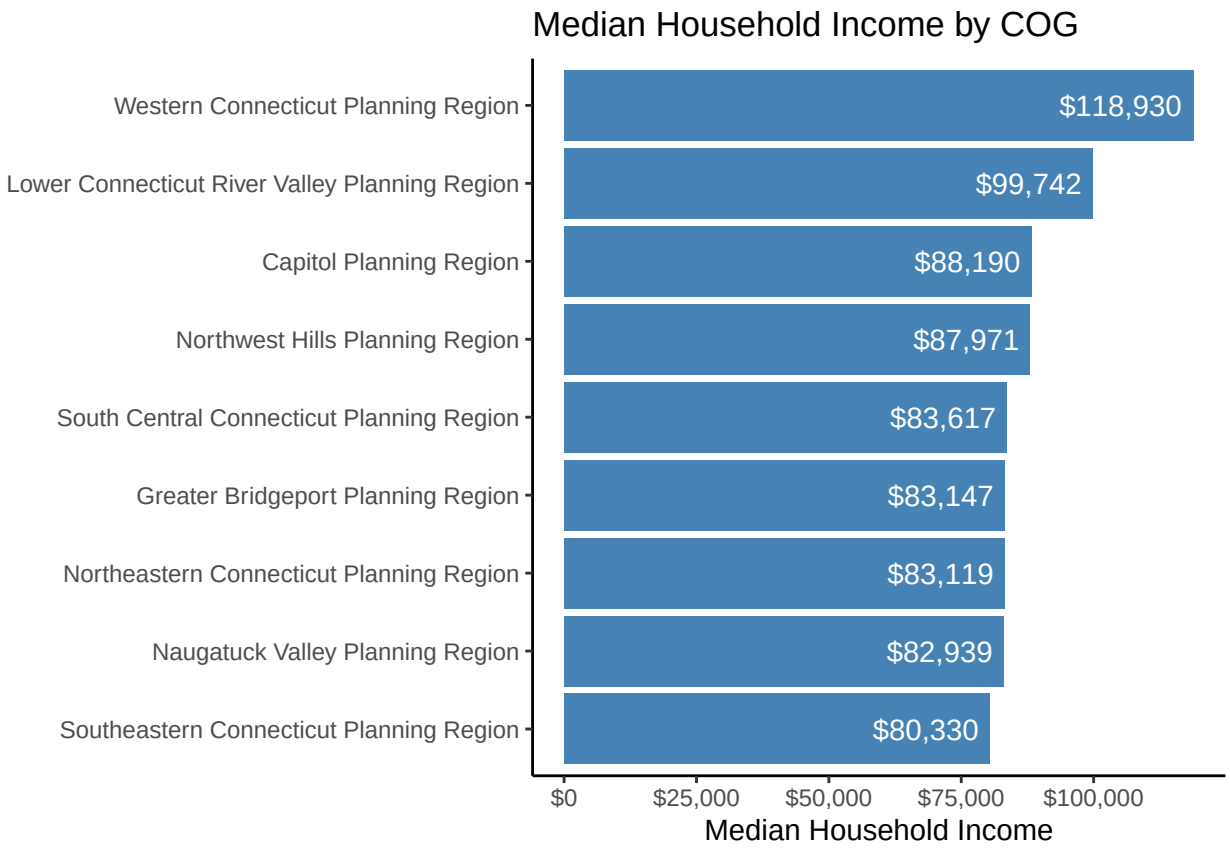
`get_acs()` defaults to the 5-year ACS, but 1-year ACS data are available using `survey = "acs1"`.

```
median_hh_income_acs <- get_acs(
  geography = "county",
  state = "Connecticut",
  variables = "B19013_001",
  year = 2022,
  survey = "acs5")
```

Here is a simple bar chart of the median household income by COG made with the ggplot2 package.

```r
#Dropping Connecticut from the name for a cleaner axis label on chart
median_hh_income_acs_V2 <- median_hh_income_acs %>%
  separate(NAME, sep = ",", c("CountyEqv", NA))

median_hh_income_acs_V2 %>%
  ggplot(aes(x = estimate, y = reorder(CountyEqv, estimate))) +
  geom_bar(stat="identity", fill="steelblue") +
  scale_x_continuous(labels = scales::dollar_format()) +
  labs(
    title = "Median Household Income by COG",
    x = "Median Household Income",
    y = NULL
    ) +
  geom_text(aes(label = scales::dollar_format()(estimate)), hjust = 1.1, colour = "white")+
  theme_classic()
```



Next, this is an example of pulling a whole table rather than specific variables. We will look at Median Age by Sex

```r
median_age_acs_table <- get_acs(
  geography = "county",
  state = "Connecticut",
  table = "B23013",
  year = 2022,
  survey = "acs5")
```

Instead of pulling the table in a long-format (the default), you can use `output = "wide"` to format the data so each variable has it's own column.

```
median_age_acs_table <- get_acs(
  geography = "county",
  state = "Connecticut",
  table = "B23013",
  year = 2022,
  survey = "acs5",
  output = "wide")
```

Now let's create a output table for Median Age by Sex. We will use the insight package to create an exportable formatted table.

```
#Dropping Connecticut from the name for a cleaner table
median_age_acs_table_v2 <- median_age_acs_table %>%
  separate(NAME, sep = ",", c("COG", NA)) %>%
  rename(`Median Age All` = B23013_001E, `Median Age All MOE` = B23013_001M,
         `Median Age Male`  = B23013_002E, `Median Age Male MOE` = B23013_002M,
         `Median Age Female`  = B23013_003E, `Median Age Female MOE` = B23013_003M) %>%
  select(!GEOID)

export_table(median_age_acs_table_v2, format = "md") #using the Insights package and markdown format
```

COG	Median Age All	Median Age All MOE	Median Age Male	Median Age Male MOE	Median Age Female	Median Age Female MOE
Capitol Planning Region	40.20	0.20	40.20	0.30	40.20	0.30
Greater Bridgeport Planning Region	42.20	0.40	42.20	0.50	42.10	0.40
Lower Connecticut River Valley Planning Region	42.60	0.30	42.40	0.50	42.90	0.40
Naugatuck Valley Planning Region	41.10	0.30	41.00	0.40	41.20	0.40
Northeastern Connecticut Planning Region	41.40	0.70	41.90	0.90	40.90	0.80
Northwest Hills Planning Region	42.90	0.60	42.60	0.60	43.30	0.80
South Central Connecticut Planning Region	40.10	0.30	40.30	0.40	39.90	0.40
Southeastern Connecticut Planning Region	40.20	0.40	39.40	0.60	41.20	0.40
Western Connecticut Planning Region	42.40	0.20	42.60	0.30	42.20	0.30