Visualizing United States Census Data

Eli Turkel
Nadia Antony

5/21/2019

Agenda

- History of the U.S. Census
- Getting started with Tidycensus
- Concepts - Tidy Data
- Step by step analysis

Follow along to create some cool maps!

US Census Timeline

- 1790: 6 question survey / U.S. pop just under 4 million
- 1810: Census adds data collection on U.S. manufacturers
- 1849: Congress establishes Census Board
- 1890: Herman Hollerith's tabulation machine used in data collection
- 1900: Census becomes a permant agency and inter-decennial data collection begins
- 1920: For first time, majority of U.S. pop lives in urban areas / U.S. pop over 106 million
- 1940: First use of statistical sampling
- 1960: DIME (Dual Independent Map Encoding)
- 1970: All data products available on magnetic computer tape
- 1980: Mailout / Mailin Surveys
- 2000: U.S. Census asks ten questions
- 2010: American Community Survey 5-year estimates released

Tidycensus Overview

- Developed by Kyle Walker
- Check out the package on Github!
- Tidycensus is an API wrapper that allows R users to ingest decennial surveys and the American Community Survey (ACS) estimates
- Census data back to the 1990 decennial survey is available
- The defaul in the 5-year ACS estimate from 2013-2017

Getting an API key

- https://api.census.gov/data/key_signup.html
- You'll then recieve an e-mail from the census with your API key

```
api_key <- "xxxxxxxx"
census_api_key(api_key, install = TRUE)
Sys.getenv("CENSUS_API_KEY")
```

Variable Search

```
library(tidycensus)
v17 <- load_variables(2017, "acs5", cache = TRUE)
v17
```

```
## # A tibble: 25,070 x 3
##    name       label                                  concept
##    <chr>      <chr>                                  <chr>
##  1 B00001_001 Estimate!!Total                        UNWEIGHTED SAMPLE COUNT OF THE POPULATION
##  2 B00002_001 Estimate!!Total                        UNWEIGHTED SAMPLE HOUSING UNITS
##  3 B01001_001 Estimate!!Total                        SEX BY AGE
##  4 B01001_002 Estimate!!Total!!Male                  SEX BY AGE
##  5 B01001_003 Estimate!!Total!!Male!!Under 5 years   SEX BY AGE
##  6 B01001_004 Estimate!!Total!!Male!!5 to 9 years    SEX BY AGE
##  7 B01001_005 Estimate!!Total!!Male!!10 to 14 years  SEX BY AGE
##  8 B01001_006 Estimate!!Total!!Male!!15 to 17 years  SEX BY AGE
##  9 B01001_007 Estimate!!Total!!Male!!18 and 19 years SEX BY AGE
## 10 B01001_008 Estimate!!Total!!Male!!20 years        SEX BY AGE
## # … with 25,060 more rows
```

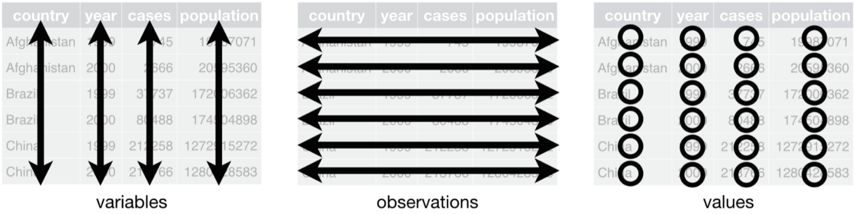
Load Var Function

```r
vt <- get_acs(geography = "county",
              variables = c(medincome = "B19013_001"),
              state = "VT")
```

What is Tidy Data?

Three interrelated rules:

1. Each variable must have its own column.
2. Each observation must have its own row.
3. Each value must have its own cell.



https://r4ds.had.co.nz/tidy-data.html#fig:tidy-structure

Why should I care?

R is a vectorized language, meaning you can do operations like:

```
v1 <- c(1, 2, 3, 4)
v2 <- c(5, 6, 7, 8)

v2 - v1
```

```
## [1] 4 4 4 4
```

instead of writing a *for loop* to subtract the individual elements

The packages inside the tidyverse, e.g `dplyr`, let you do data cleaning and manipulation operations easily when data is in tidy format.

Using these packages can help - write faster and more 'readable' code

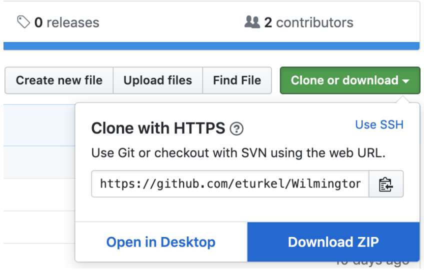
Brief concepts - commands you'll see

- **`%>%`** the “pipe” operator for chaining
- **`filter`** to subset a dataframe
- **`group_by`** and then **`summarise`**
- **`facet`** to create *small multiples* plots
- **`left_join`** to join datasets
- and **`<-`** is the assignment operator, more explicit than =

Let's Start the Analysis!

Setup

- Install R and RStudio
- Download the Repository



Clone with HTTPS

Use Git or checkout with SVN using the web URL.

https://github.com/eturkel/Wilmingtor

OR

- Use this RStudio Cloud workspace: https://rstudio.cloud/project/355872

Package Dependencies

Install packages (if local setup)...

```r
install.packages("sf")
install.packages("tidycensus")
install.packages("dplyr")
install.packages("ggplot2")
install.packages("tidyr")
install.packages("purrr")
install.packages("lwgeom")
```

... and load libraries

```r
library(dplyr)
library(ggplot2)
library(tidyr)
library(sf)
library(tidycensus)
library(tigris)
library(purrr)
```

Your Choices!

Get an API Key from http://api.census.gov/data/key_signup.html

```
census_api_key("<YOUR API KEY>")
demo_variables <- # define the variables you want to analyze here
de_census_data <- get_acs(geography = "tract",
                          state = "DE",
                          variables = demo_variables,
                          geometry = TRUE,
                          cb = TRUE)
```

OR

Load de_census_data.RData

```
load("data/de_census_data.RData")
```

Look at the data

```
head(de_census_data)
```

```
## Simple feature collection with 6 features and 5 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -75.7601 ymin: 39.17347 xmax: -75.60411 ymax: 39.29937
## epsg (SRID):    4269
## proj4string:    +proj=longlat +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +no_defs
##         GEOID                                    NAME             variable estimate moe
## 1 10001040100 Census Tract 401, Kent County, Delaware                white     6080 502
## 2 10001040100 Census Tract 401, Kent County, Delaware                black      501 100
## 3 10001040100 Census Tract 401, Kent County, Delaware                asian       58  60
## 4 10001040100 Census Tract 401, Kent County, Delaware             hispanic      265 198
## 5 10001040100 Census Tract 401, Kent County, Delaware          foreignborn      132 125
## 6 10001040100 Census Tract 401, Kent County, Delaware high_school_diplomas     1808 273
##                          geometry
## 1 MULTIPOLYGON (((-75.7601 39...
## 2 MULTIPOLYGON (((-75.7601 39...
## 3 MULTIPOLYGON (((-75.7601 39...
## 4 MULTIPOLYGON (((-75.7601 39...
## 5 MULTIPOLYGON (((-75.7601 39...
## 6 MULTIPOLYGON (((-75.7601 39...
```

Some cleaning using %>%

```r
de_census_data_clean <- de_census_data %>%
    separate(col = NAME,
             into = c("Census_Tract", "County", "State"),
             sep = ",") %>%
    separate(col = Census_Tract,
             into = c(NA, NA, "Census_Tract_Number"),
             sep = " ")
```

Using the grammar

Let's read the previous code step by step

```
de_census_data %>% # take the data, and then
    separate(col = NAME,
             into = c("Census_Tract", "County", "State"),
             sep = ",") # separate by ','
```

```
## Simple feature collection with 2616 features and 7 fields (with 12 geometries empty)
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -75.78866 ymin: 38.45101 xmax: -75.04894 ymax: 39.83901
## epsg (SRID):    4269
## proj4string:    +proj=longlat +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +no_defs
## First 10 features:
##          GEOID     Census_Tract       County     State                     variable estimate  moe
## 1  10001040100 Census Tract 401  Kent County  Delaware                        white     6080  502
## 2  10001040100 Census Tract 401  Kent County  Delaware                        black      501  100
## 3  10001040100 Census Tract 401  Kent County  Delaware                        asian       58   60
## 4  10001040100 Census Tract 401  Kent County  Delaware                     hispanic      265  198
## 5  10001040100 Census Tract 401  Kent County  Delaware                  foreignborn      132  125
## 6  10001040100 Census Tract 401  Kent County  Delaware         high_school_diplomas     1808  273
## 7  10001040100 Census Tract 401  Kent County  Delaware             bachelor_degrees      268  122
## 8  10001040100 Census Tract 401  Kent County  Delaware              masters_degrees      181   97
## 9  10001040100 Census Tract 401  Kent County  Delaware households_earning_over_200k       51   50
## 10 10001040100 Census Tract 401  Kent County  Delaware                median_income    63324 8985
##                          geometry
## 1  MULTIPOLYGON (((-75.7601 39...
## 2  MULTIPOLYGON (((-75.7601 39...
## 3  MULTIPOLYGON (((-75.7601 39...
## 4  MULTIPOLYGON (((-75.7601 39...
## 5  MULTIPOLYGON (((-75.7601 39...
## 6  MULTIPOLYGON (((-75.7601 39...
## 7  MULTIPOLYGON (((-75.7601 39...
## 8  MULTIPOLYGON (((-75.7601 39...
## 9  MULTIPOLYGON (((-75.7601 39...
## 10 MULTIPOLYGON (((-75.7601 39...
```

Using the grammar (continued)

```
de_census_data %>% # take the data, and then
    separate(col = NAME,
             into = c("Census_Tract", "County", "State"),
             sep = ",") %>% # separate, and then
    separate(col = Census_Tract,
             into = c(NA, NA, "Census_Tract_Number"),
             sep = " ") # separate out Number
```

```
## Simple feature collection with 2616 features and 7 fields (with 12 geometries empty)
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -75.78866 ymin: 38.45101 xmax: -75.04894 ymax: 39.83901
## epsg (SRID):    4269
## proj4string:    +proj=longlat +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +no_defs
## First 10 features:
##          GEOID Census_Tract_Number       County    State                     variable estimate
## 1  10001040100                 401  Kent County  Delaware                        white     6080
## 2  10001040100                 401  Kent County  Delaware                        black      501
## 3  10001040100                 401  Kent County  Delaware                        asian       58
## 4  10001040100                 401  Kent County  Delaware                     hispanic      265
## 5  10001040100                 401  Kent County  Delaware                  foreignborn      132
## 6  10001040100                 401  Kent County  Delaware         high_school_diplomas     1808
## 7  10001040100                 401  Kent County  Delaware             bachelor_degrees      268
## 8  10001040100                 401  Kent County  Delaware              masters_degrees      181
## 9  10001040100                 401  Kent County  Delaware households_earning_over_200k       51
## 10 10001040100                 401  Kent County  Delaware                median_income    63324
##     moe                       geometry
## 1   502 MULTIPOLYGON (((-75.7601 39...
## 2   100 MULTIPOLYGON (((-75.7601 39...
## 3    60 MULTIPOLYGON (((-75.7601 39...
## 4   198 MULTIPOLYGON (((-75.7601 39...
## 5   125 MULTIPOLYGON (((-75.7601 39...
## 6   273 MULTIPOLYGON (((-75.7601 39...
## 7   122 MULTIPOLYGON (((-75.7601 39...
## 8    97 MULTIPOLYGON (((-75.7601 39...
## 9    50 MULTIPOLYGON (((-75.7601 39...
## 10 8985 MULTIPOLYGON (((-75.7601 39...
```

Creating the Map!

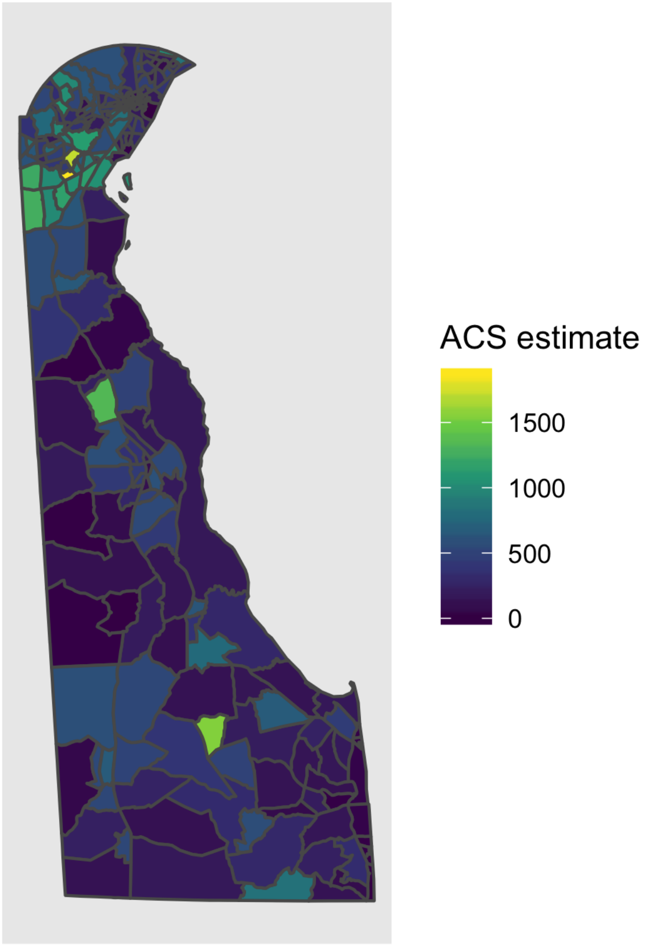
First pick one variable to view estimates

```
# Let's look at the number of foreignborn in each Tract
## create a data frame by subsetting only the 'foreignborn'
de_census_fb <- de_census_data_clean %>%
  filter(variable %in% c("foreignborn"))
```

Now let's plot this:

```
ggplot(de_census_fb, aes(fill = estimate)) +
  geom_sf() +
  scale_fill_viridis_c() +
  coord_sf(datum = NA) +
  labs(title = "Foreign-Born Estimates by DE Census Tract",
       caption = "Data: 2013-2017 5-year ACS",
       fill = "ACS estimate")
```

Foreign-Born Estimates by DE Census Tract



Data: 2013-2017 5-year ACS

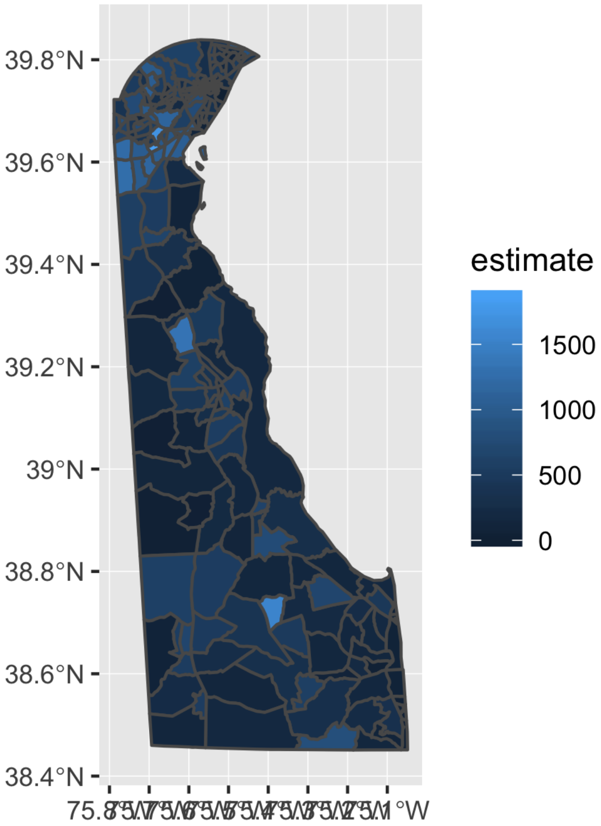
See the individual plot layers

We use the ggplot2 package for **layering** plot info. `geom_sf` is used to map the varied shapes (polygons, lines)

```
ggplot(de_census_fb, aes(fill = estimate))
```

Add the geometries

```
ggplot(de_census_fb, aes(fill = estimate)) +
  geom_sf()
```

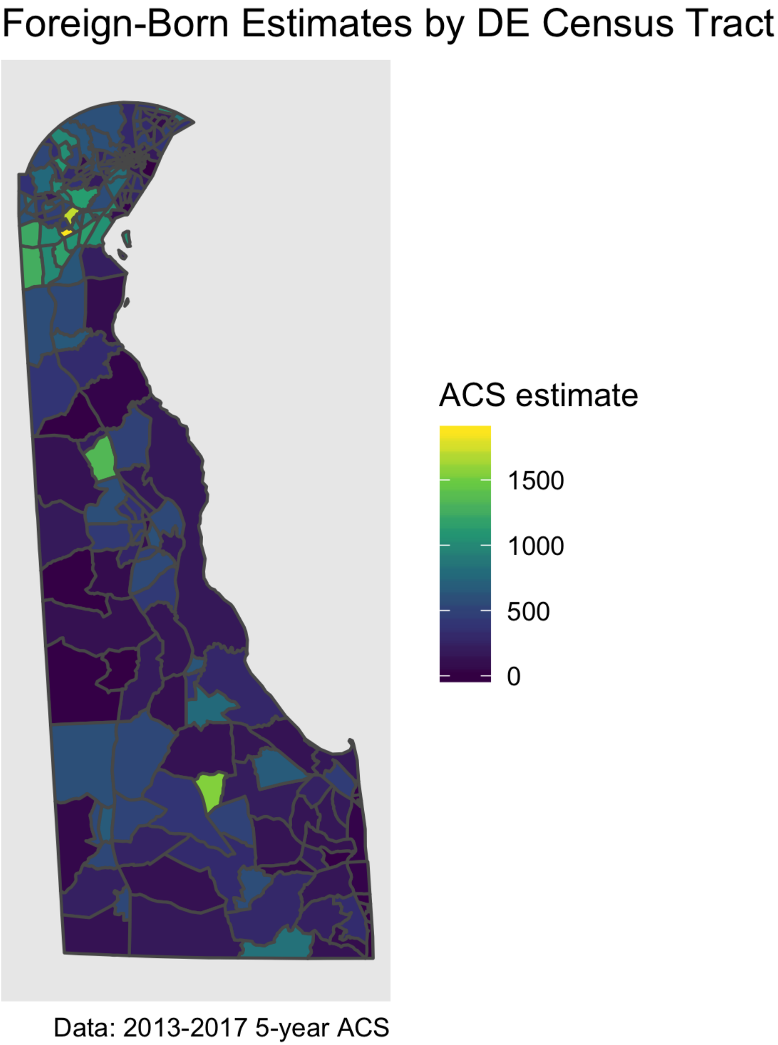


Add safe colors

```
ggplot(de_census_fb, aes(fill = estimate)) +
  geom_sf() +
  scale_fill_viridis_c()
```

Add the theme and title

```
ggplot(de_census_fb, aes(fill = estimate)) +
  geom_sf() +
  scale_fill_viridis_c() +
  coord_sf(crs = 26916, datum = NA) +
  labs(title = "Foreign-Born Estimates by DE Census Tract",
       caption = "Data: 2013-2017 5-year ACS",
       fill = "ACS estimate")
```


Foreign-Born Estimates by DE Census Tract
ACS estimate
1500
1000
500
0
Data: 2013-2017 5-year ACS

Subset only Wilmington areas using Tract

```
# Create a dataframe with only Wilmington Tracts
wilm_census_data <- de_census_data_clean %>%
  filter(Census_Tract_Number %in% c(2, 3, 4, 5, 6.01, 6.02,
                                    9, 11, 12, 13, 14, 15,
                                    16, 19.02, 21, 22, 23, 24,
                                    25, 26, 27, 28, 29, 30.02))
```

Cleaning came of help here to filter out only the relevant numbers

Let's plot EVERYTHING!

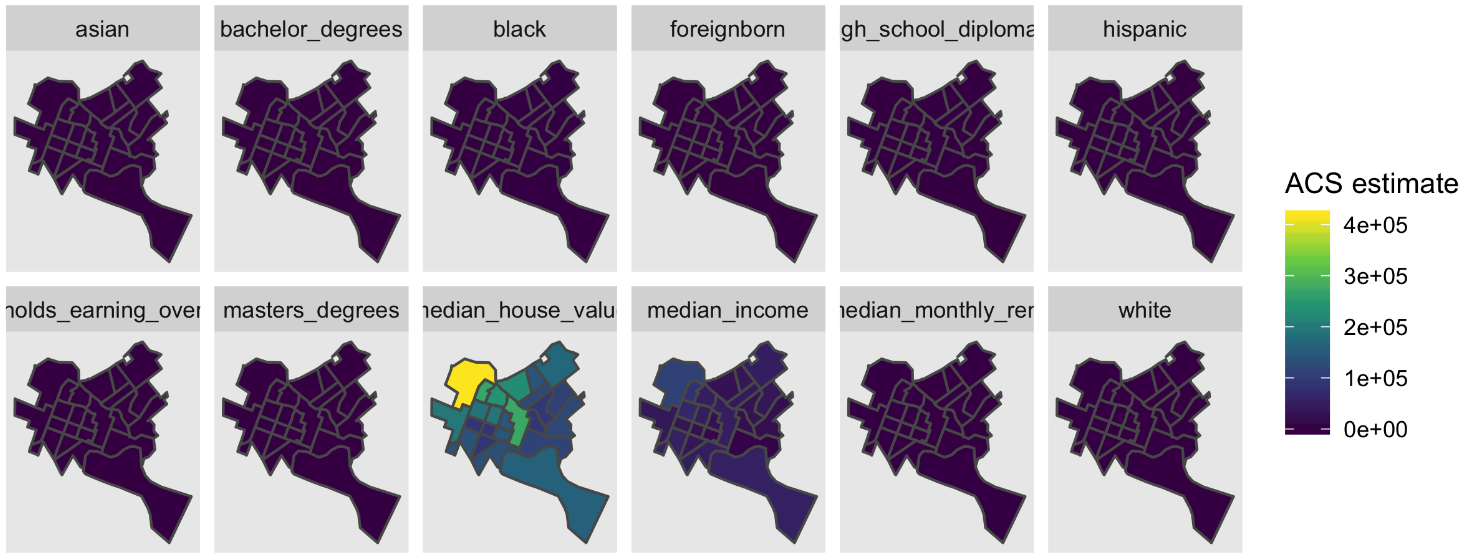
```
# Plot all our data
ggplot(wilm_census_data, aes(fill = estimate)) +
  geom_sf() +
  scale_fill_viridis_c() +
  coord_sf(crs = 26916, datum = NA) +
  labs(title = "Estimates by Census Tract",
       subtitle = "Wilmington, DE",
       caption = "Data: 2013-2017 5-year ACS
       \nData acquired with the R tidycensus package.",
       fill = "ACS estimate") +
  facet_wrap(~variable)
```

`facet` helps you split up the data by variable and plot each

What's the problem?

Estimates by Census Tract

Wilmington, DE



Data: 2013-2017 5-year ACS

Data acquired with the R tidycensus package.

Plotting some comparable variables

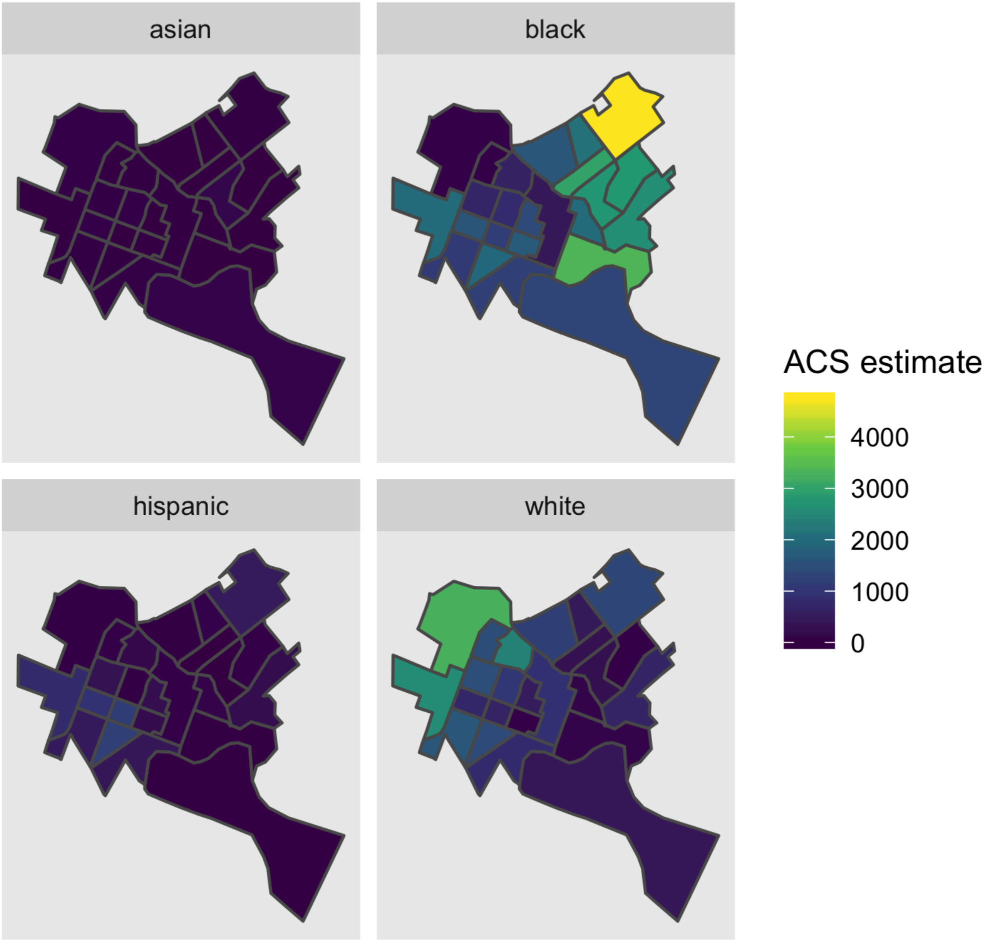
Let's focus on the race variables

```
## create a data frame with race variables
wilm_census_race <- wilm_census_data %>%
  filter(variable %in% c("hispanic", "black", "asian", "white"))
```

```
## plot
ggplot(wilm_census_race, aes(fill = estimate)) +
  geom_sf() +
  scale_fill_viridis_c() +
  coord_sf(crs = 26916, datum = NA) +
  labs(title = "Population Estimates",
       subtitle = "Wilmington, DE",
       fill = "ACS estimate") +
  facet_wrap(~variable)
```

Population Estimates

Wilmington, DE



Comparing a better way

'Small multiples plots' are useful to compare between variables. But we need to make sure we compare the right proportions so as to not let people take away a wrong insight.

Let's do some data aggregation and data joins to find the percentages within each tract.

Code

Create the total population data frame

```r
## Estimate the total population
## by summing up the different race estimates
wilm_tract_pop <- wilm_census_race %>%
  group_by(Census_Tract_Number) %>%
  summarise(Population_Estimate = sum(estimate))
```

```r
# remove geometry variable to make it a regular dataset
st_geometry(wilm_tract_pop) <- NULL
```

Join the total population data with the original and create the Percentage column

```r
# create data frame with the percentages
wilm_tract_percpop <- wilm_census_race %>%
  left_join(wilm_tract_pop, by = "Census_Tract_Number") %>%
  mutate(Percentage = estimate/Population_Estimate)
```

Step by Step

Highlight sections and run using `cmd` + `return` to see the separate steps. Remember not to highlight the variable assignment part.

```
wilm_census_race %>%
  group_by(Census_Tract_Number)
```

```
## Simple feature collection with 96 features and 7 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -75.5885 ymin: 39.7005 xmax: -75.51268 ymax: 39.77263
## epsg (SRID):    4269
## proj4string:    +proj=longlat +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +no_defs
## # A tibble: 96 x 8
## # Groups:   Census_Tract_Number [24]
##    GEOID  Census_Tract_Num… County   State variable estimate   moe                          geometry
##    <chr>  <chr>             <chr>    <chr> <chr>       <dbl> <dbl>                <MULTIPOLYGON [°]>
##  1 10003… 2                 " New C… " De… white        1343   288 (((-75.53885 39.76623, -75.53547…
##  2 10003… 2                 " New C… " De… black        4741   722 (((-75.53885 39.76623, -75.53547…
##  3 10003… 2                 " New C… " De… asian          32    60 (((-75.53885 39.76623, -75.53547…
##  4 10003… 2                 " New C… " De… hispanic      505   342 (((-75.53885 39.76623, -75.53547…
##  5 10003… 3                 " New C… " De… white         468   133 (((-75.54275 39.76461, -75.53923…
##  6 10003… 3                 " New C… " De… black        2111   244 (((-75.54275 39.76461, -75.53923…
##  7 10003… 3                 " New C… " De… asian           0    11 (((-75.54275 39.76461, -75.53923…
##  8 10003… 3                 " New C… " De… hispanic      147   121 (((-75.54275 39.76461, -75.53923…
##  9 10003… 4                 " New C… " De… white        1197   209 (((-75.55714 39.76163, -75.55692…
## 10 10003… 4                 " New C… " De… black        1645   291 (((-75.55714 39.76163, -75.55692…
## # … with 86 more rows
```

```r
wilm_census_race %>%
  group_by(Census_Tract_Number) %>%
  summarise(Population_Estimate = sum(estimate))
```

```
## Simple feature collection with 24 features and 2 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -75.5885 ymin: 39.7005 xmax: -75.51268 ymax: 39.77263
## epsg (SRID):    4269
## proj4string:    +proj=longlat +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +no_defs
## # A tibble: 24 x 3
##    Census_Tract_Numb… Population_Estima…                                                    geometry
##    <chr>                           <dbl>                                          <MULTIPOLYGON [°]>
##  1 11                               3267 (((-75.56156 39.75696, -75.56049 39.75772, -75.56151 39.75…
##  2 12                               1691 (((-75.56816 39.75632, -75.56531 39.76056, -75.56244 39.76…
##  3 13                               3508 (((-75.57996 39.76337, -75.57785 39.76644, -75.5768 39.768…
##  4 14                               2601 (((-75.57097 39.75227, -75.57013 39.75353, -75.56863 39.75…
##  5 15                               2017 (((-75.56376 39.7493, -75.56289 39.75057, -75.56229 39.751…
##  6 16                               2223 (((-75.55931 39.74559, -75.55808 39.747, -75.55719 39.7482…
##  7 19.02                            1879 (((-75.56028 39.73048, -75.55854 39.73517, -75.55634 39.73…
##  8 2                                6621 (((-75.53885 39.76623, -75.53547 39.76727, -75.53466 39.76…
##  9 21                               2044 (((-75.56197 39.74169, -75.56072 39.74348, -75.56032 39.74…
## 10 22                               2990 (((-75.5662 39.74573, -75.56479 39.74781, -75.55931 39.745…
## # … with 14 more rows
```

```
wilm_census_race %>%
  left_join(wilm_tract_pop, by = "Census_Tract_Number")
```

```
## Simple feature collection with 96 features and 8 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -75.5885 ymin: 39.7005 xmax: -75.51268 ymax: 39.77263
## epsg (SRID):    4269
## proj4string:    +proj=longlat +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +no_defs
## First 10 features:
##          GEOID Census_Tract_Number             County     State variable estimate moe
## 1  10003000200                   2  New Castle County  Delaware    white     1343 288
## 2  10003000200                   2  New Castle County  Delaware    black     4741 722
## 3  10003000200                   2  New Castle County  Delaware    asian       32  60
## 4  10003000200                   2  New Castle County  Delaware hispanic      505 342
## 5  10003000300                   3  New Castle County  Delaware    white      468 133
## 6  10003000300                   3  New Castle County  Delaware    black     2111 244
## 7  10003000300                   3  New Castle County  Delaware    asian        0  11
## 8  10003000300                   3  New Castle County  Delaware hispanic      147 121
## 9  10003000400                   4  New Castle County  Delaware    white     1197 209
## 10 10003000400                   4  New Castle County  Delaware    black     1645 291
##    Population_Estimate                       geometry
## 1                 6621 MULTIPOLYGON (((-75.53885 3...
## 2                 6621 MULTIPOLYGON (((-75.53885 3...
## 3                 6621 MULTIPOLYGON (((-75.53885 3...
## 4                 6621 MULTIPOLYGON (((-75.53885 3...
## 5                 2726 MULTIPOLYGON (((-75.54275 3...
## 6                 2726 MULTIPOLYGON (((-75.54275 3...
## 7                 2726 MULTIPOLYGON (((-75.54275 3...
## 8                 2726 MULTIPOLYGON (((-75.54275 3...
## 9                 2890 MULTIPOLYGON (((-75.55714 3...
## 10                2890 MULTIPOLYGON (((-75.55714 3...
```

```r
wilm_census_race %>%
  left_join(wilm_tract_pop, by = "Census_Tract_Number") %>%
  mutate(Percentage = estimate/Population_Estimate)
```

```
## Simple feature collection with 96 features and 9 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -75.5885 ymin: 39.7005 xmax: -75.51268 ymax: 39.77263
## epsg (SRID):    4269
## proj4string:    +proj=longlat +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +no_defs
## First 10 features:
##         GEOID Census_Tract_Number             County    State variable estimate moe
## 1  10003000200                   2  New Castle County  Delaware    white     1343 288
## 2  10003000200                   2  New Castle County  Delaware    black     4741 722
## 3  10003000200                   2  New Castle County  Delaware    asian       32  60
## 4  10003000200                   2  New Castle County  Delaware hispanic      505 342
## 5  10003000300                   3  New Castle County  Delaware    white      468 133
## 6  10003000300                   3  New Castle County  Delaware    black     2111 244
## 7  10003000300                   3  New Castle County  Delaware    asian        0  11
## 8  10003000300                   3  New Castle County  Delaware hispanic      147 121
## 9  10003000400                   4  New Castle County  Delaware    white     1197 209
## 10 10003000400                   4  New Castle County  Delaware    black     1645 291
##    Population_Estimate                       geometry  Percentage
## 1                 6621 MULTIPOLYGON (((-75.53885 3... 0.202839450
## 2                 6621 MULTIPOLYGON (((-75.53885 3... 0.716054977
## 3                 6621 MULTIPOLYGON (((-75.53885 3... 0.004833107
## 4                 6621 MULTIPOLYGON (((-75.53885 3... 0.076272466
## 5                 2726 MULTIPOLYGON (((-75.54275 3... 0.171680117
## 6                 2726 MULTIPOLYGON (((-75.54275 3... 0.774394718
## 7                 2726 MULTIPOLYGON (((-75.54275 3... 0.000000000
## 8                 2726 MULTIPOLYGON (((-75.54275 3... 0.053925165
## 9                 2890 MULTIPOLYGON (((-75.55714 3... 0.414186851
## 10                2890 MULTIPOLYGON (((-75.55714 3... 0.569204152
```

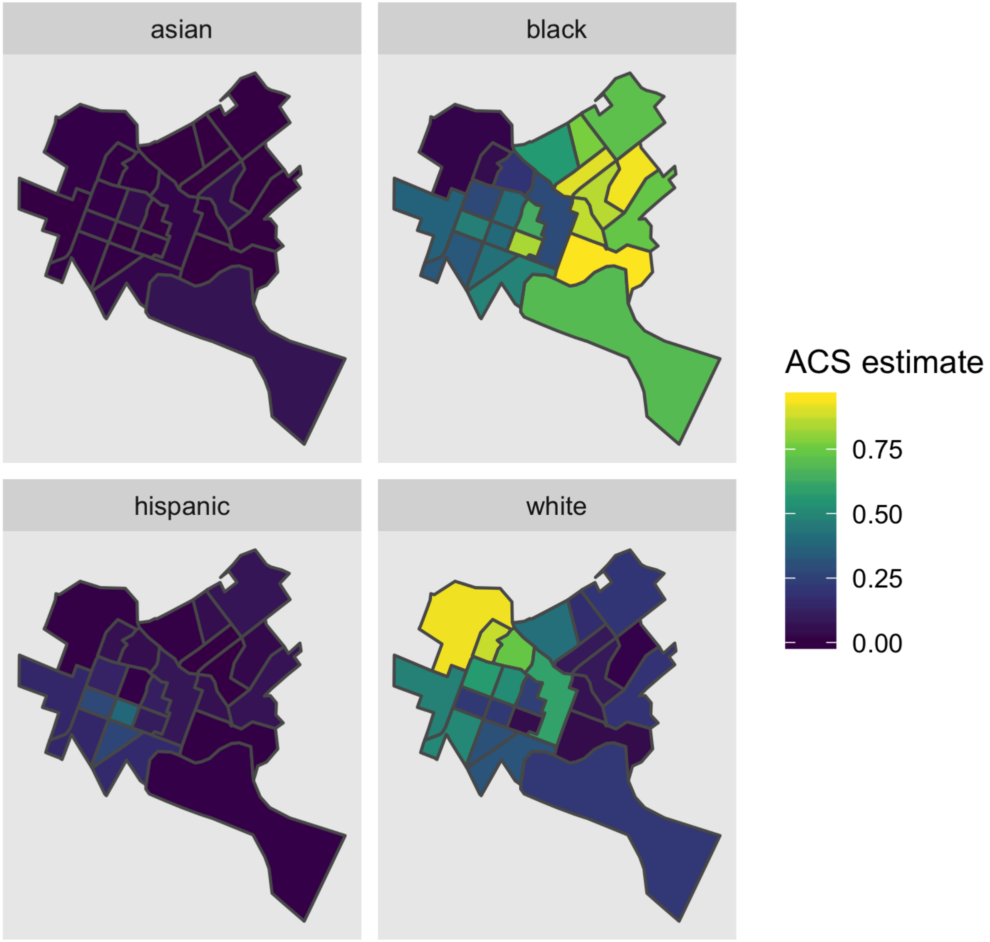
Note: `tidycensus` allows you to get the summary value through the API as well!

Plot the percentage

```r
ggplot(wilm_tract_percpop, aes(fill = Percentage)) +
  geom_sf() +
  scale_fill_viridis_c() +
  coord_sf(crs = 26916, datum = NA) +
  labs(title = "Percentage of Total (Estimates) by Census Tract",
       subtitle = "Wilmington, DE",
       fill = "ACS estimate") +
  facet_wrap(~variable)
```

Percentage of Total (Estimates) by Census Tract

Wilmington, DE



Your Turn

You can create a Wilmington Education variable by filtering `c("high_school_diplomas", "bachelor_degrees", "masters_degrees")` variables and recreating the previous visualizations:

- Count estimates by census tract
- Percentage estimates by census tract

Answer

```
wilm_census_edu <- wilm_census_data %>%
  filter(variable %in% c("high_school_diplomas",
                         "bachelor_degrees",
                         "masters_degrees"))

ggplot(wilm_census_edu, aes(fill = estimate)) +
  geom_sf() +
  scale_fill_viridis_c() +
  coord_sf(crs = 26916, datum = NA) +
  labs(title = "Education Estimates",
       subtitle = "Wilmington, DE",
       fill = "ACS estimate") +
  facet_wrap(~variable)
```

```r
# As a percentage of education data available
# (or use total population)
wilm_tract_totedu <- wilm_census_edu %>%
  group_by(Census_Tract_Number) %>%
  summarise(Education_Estimate = sum(estimate))

st_geometry(wilm_tract_totedu) <- NULL # remove geometry

wilm_tract_percedu <- wilm_tract_totedu %>%
  left_join(wilm_census_edu, by = "Census_Tract_Number") %>%
  mutate(Percentage = estimate/Education_Estimate)

# Plot
ggplot(wilm_tract_percedu, aes(fill = Percentage)) +
  geom_sf() +
  scale_fill_viridis_c() +
  coord_sf(crs = 26916, datum = NA) +
  labs(title = "Percentage of Education Estimates by Census Tract",
       subtitle = "Wilmington, DE",
       fill = "ACS estimate") +
  facet_wrap(~variable)
```

Dot Density Plots

Choropleth maps have a tendency of being misunderstood due to the *area* covered by a color. We can plot dots in order to avoid the issue of misrepresentation of sparsely populated areas and give an idea of density.

Hold tight as this will have some heavy lifting with functions from dplyr and purrr!

```
wm_dots <- map(c("white", "black",
                 "asian", "hispanic"), function(group) {
    wilm_census_data %>%
        filter(variable == group) %>%
        st_sample(., size = .$estimate / 10, exact = FALSE) %>%
        st_sf() %>%
        mutate(group = group)
}) %>%
    reduce(rbind) %>%
    group_by(group) %>%
    summarize()
```

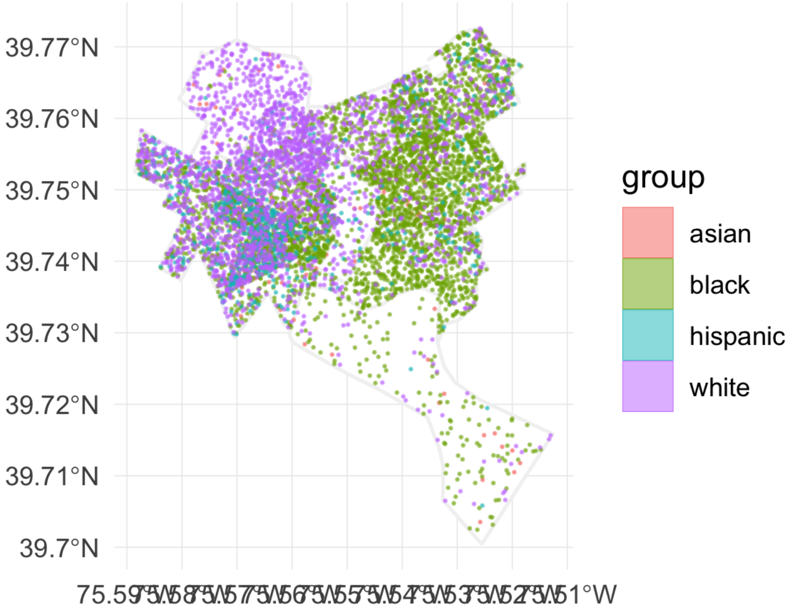
`map()` Applys a function to each element of a vector, in our case the vector is the race values. i.e, for each race we subset the wilmington data, and create dots representing the population in each tract.

```r
# for each group
group <- "asian"
wilm_census_data %>%
        filter(variable == group) %>%
        st_sample(., size = .$estimate / 10, exact = FALSE) %>%
        st_sf() %>%
        mutate(group = group)
```

- `st_sample()` generates a sample of random dots each one representing 10 people.
- `st_sf()` converts the POINT geometry set back to simple features dataframe.

Plotting it all together

```
ggplot() +
    geom_sf(data = wilm_census_data, color = "grey95", fill = "white'
    geom_sf(data = wm_dots, aes(color = group, fill = group),
            size = 0.1, alpha = 0.5) +
    theme_minimal()
```



Looking back

Data being tidy allowed us to immediately use commands like:

- facet
- group_by

Remember ACS are estimates so we should consider the MOE or Margin of Error variable.

Next steps

- Try using another layer with alpha to show the MOE *
- Integrate this data with statistics from the Uniform Crime Reporting database
- Ask interesting questions and use tidy functions to get quick results
    - What are the high median rent areas?
    - Does the median house value correlate with houses earning above 200k?
    - Among the high median house value ones what is the percentage of owners to renter?
    - Does more Education level imply more Median income?
    - Where are the Vacant houses and what is the median income in these areas?

Thank You!