Census Data, Leaflet, and more Choropleths

Data Wrangling and Husbandry

03/30/2020

The `choroplethr` package works well, but you

1. are likely to want data beyond the examples in the package
2. might want to integrate the maps into `ggplot2`
3. might want to create `leaflet` maps

Census Data

- The United States Bureau of the Census
  - performs the US Census every 10 years
  - runs many other survey and censuses, of which the most important is the American Community Survey
- To work with census data, you will want an API key
  - https://api.census.gov/data/key_signup.html
- Within R, the `acs` package or the `tidycensus` make working with census data much easier.

You can save your api key within a session using this format

```
library(tidycensus)
library(tidyverse)

census_api_key("YOUR API KEY GOES HERE")

## To install your API key for use in future sessions, run
```

- The ACS uses coded names for variables, and they change over time.
- The most recent ACS aggregated over 5 years is 2016.
  - You can look up variable names at

`https://www.socialexplorer.com/data/ACS2016_5yr/metadat`

(and many other places)
  - You can use the `load_variables()` function in the `tidycensus` package as well, although I'm not certain it gets all of the variables.

```
v15 <- load_variables(2016, "acs5", cache = TRUE)
View(v15)
```

- You'll notice that the ACS includes both estimates and margins of error; the variable names end in `E` or `M`. When we download variables using `get_acs()` we drop the letter at the end.

Median household income example

```
nj_median_household_income <-
  get_acs(geography = "county",
                variables = c(medincome = "B19013_001"),
                state = "NJ")
```

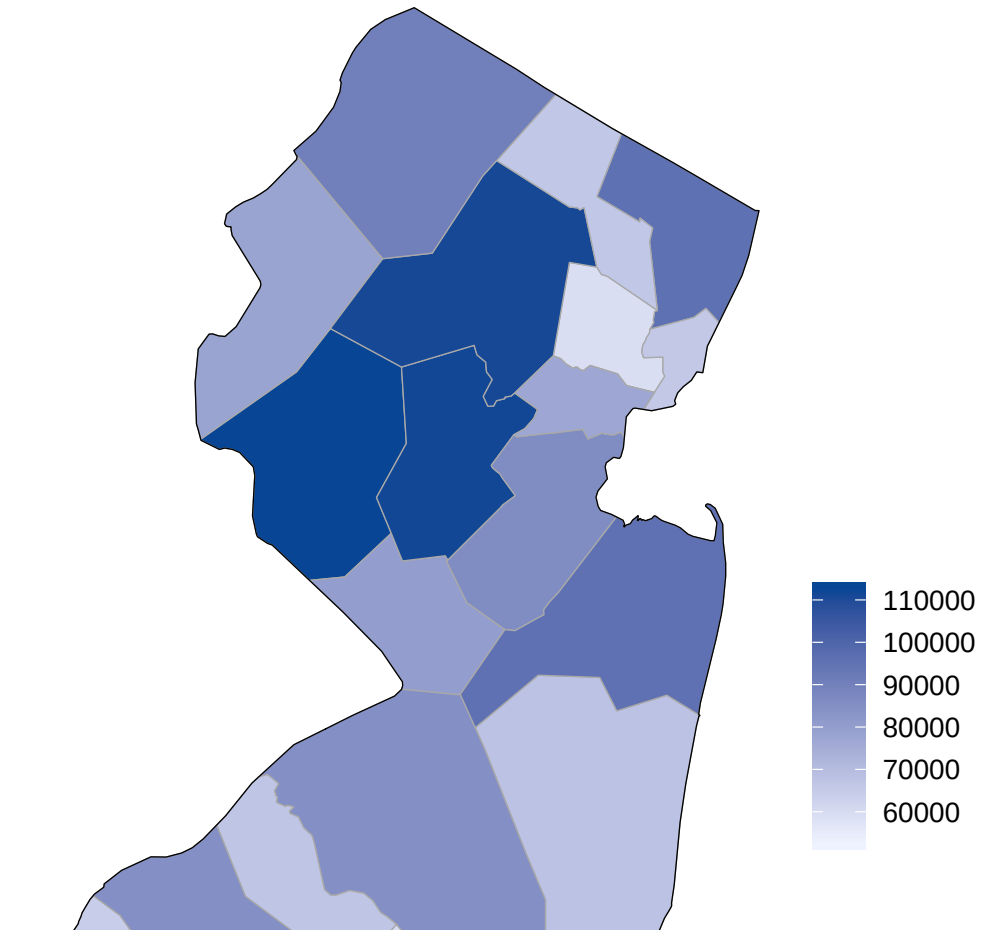
```
## Getting data from the 2014-2018 5-year ACS
```

```
glimpse(nj_median_household_income)
```

```
## Observations: 21
## Variables: 5
## $ GEOID    <chr> "34001", "34003", "34005", "34007", "34
## $ NAME     <chr> "Atlantic County, New Jersey", "Bergen
## $ variable <chr> "medincome", "medincome", "medincome",
## $ estimate <dbl> 59989, 95837, 84992, 67118, 63690, 525
## $ moe      <dbl> 1631, 998, 1319, 1282, 1641, 1718, 106
```

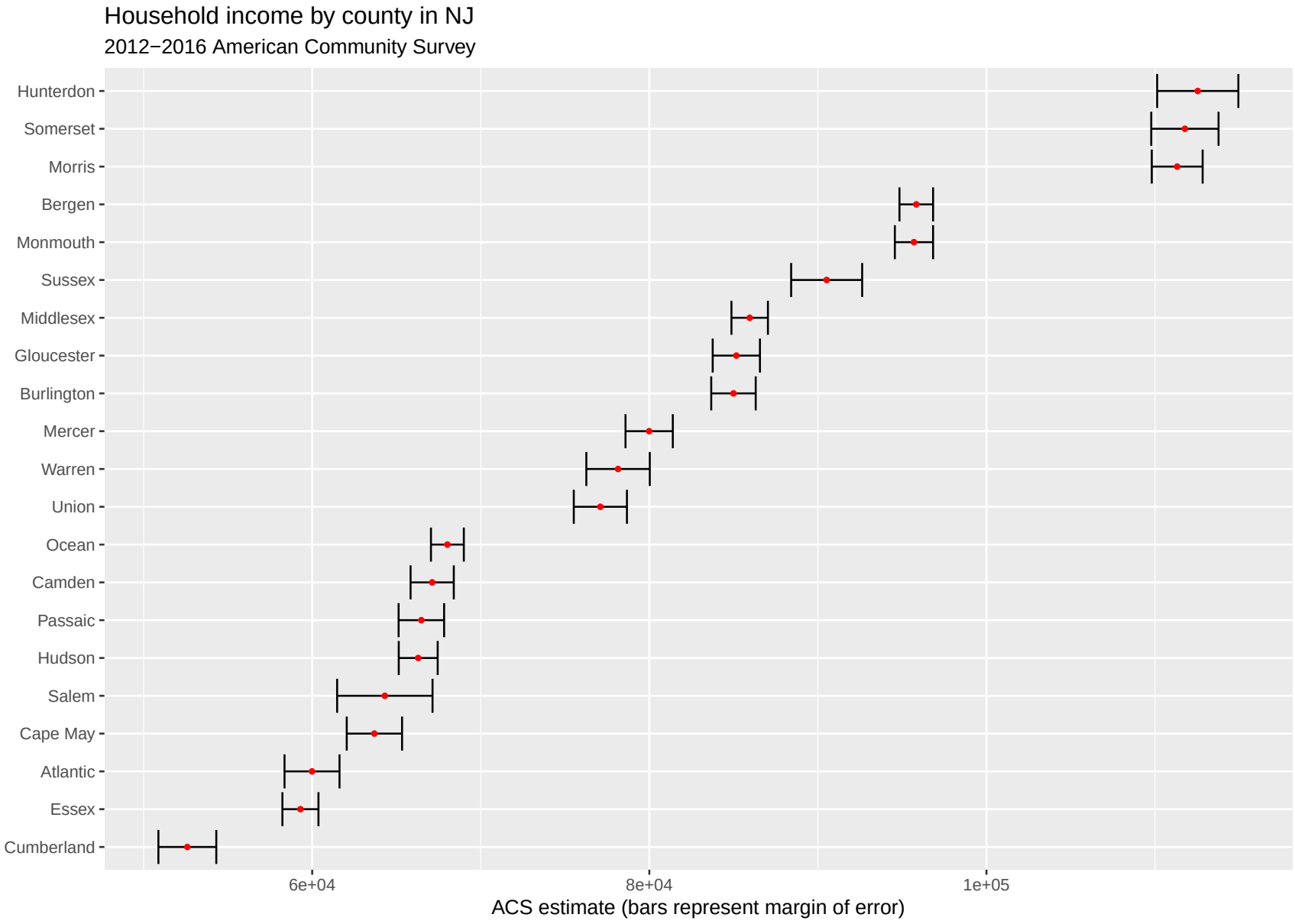
We could pipe this to the function from the `cloroplethr` package:

```
nj_median_household_income %>%
  dplyr::rename(value = estimate) %>%
  mutate(region = as.integer(GEOID)) %>%
  choroplethr::county_choropleth(title = "Median Household
```



Not every use of census data has to be a map.

```
nj_median_household_income %>%
  mutate(NAME = str_replace(NAME, " County, New Jersey", ""
  mutate(NAME = reorder(NAME, estimate)) %>%
  ggplot(aes(x = estimate, y = NAME)) +
  geom_errorbarh(aes(xmin = estimate - moe, xmax = estimate
  geom_point(color = "red", size = 1) +
  labs(title = "Household income by county in NJ",
       subtitle = "2012-2016 American Community Survey",
       y = "",
       x = "ACS estimate (bars represent margin of error)")
```


Household income by county in NJ
2012−2016 American Community Survey
Hunterdon
Somerset
Morris
Bergen
Monmouth
Sussex
Middlesex
Gloucester
Burlington
Mercer
Warren
Union
Ocean
Camden
Passaic
Hudson
Salem
Cape May
Atlantic
Essex
Cumberland
6e+04
8e+04
1e+05
ACS estimate (bars represent margin of error)

- ACS data is reported on finer geographic units than states and counties, known as "tracts":

```
## Change geography to "tract"
## and add geometry = TRUE for plotting purposes

nj_median_household_income2 <-
  get_acs(geography = "tract",
                variables = c(medincome = "B19013_001"),
                state = "NJ",
                geometry = TRUE)
```

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

## Downloading feature geometry from the Census website.  T

##   |
```

```r
glimpse(nj_median_household_income2)
```

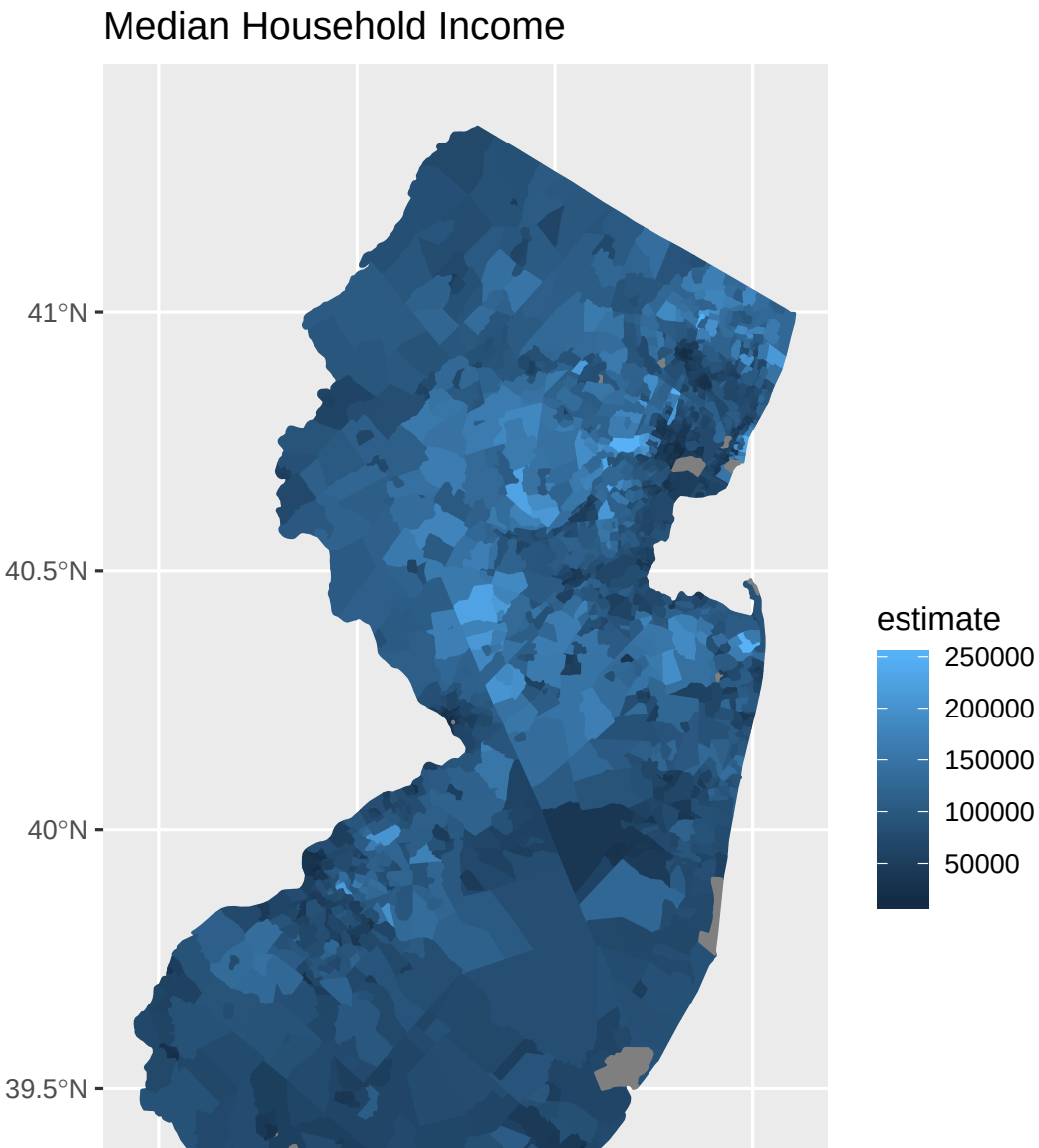
```
## Observations: 2,010
## Variables: 6
## $ GEOID    <chr> "34001000100", "34001000200", "3400100
## $ NAME     <chr> "Census Tract 1, Atlantic County, New
## $ variable <chr> "medincome", "medincome", "medincome",
## $ estimate <dbl> 34844, 46742, 36827, 31115, 38054, 146
## $ moe      <dbl> 3888, 18150, 6976, 9120, 4308, 2249, 6
## $ geometry <MULTIPOLYGON [°]> MULTIPOLYGON (((-74.46816
```

Plotting census data with ggplot2

- The `geometry = TRUE` argument in `get_acs()` downloads polygon information for plotting
- the `geom_sf()` function in `ggplot2` can use that information, but currently only the development version of the `ggplot2` package:

```
devtools::install_github("tidyverse/ggplot2")
```

```
nj_median_household_income2 %>%
  ggplot(aes(fill = estimate, color = estimate)) +
  geom_sf() +
  ggtitle("Median Household Income")
```



```
middlesex_rental <-
  get_acs(
  geography = "tract",
  variables = c(medrental = "B25064_001"),
  state = "NJ",
  county = "Middlesex",
  geometry = TRUE
  )
```

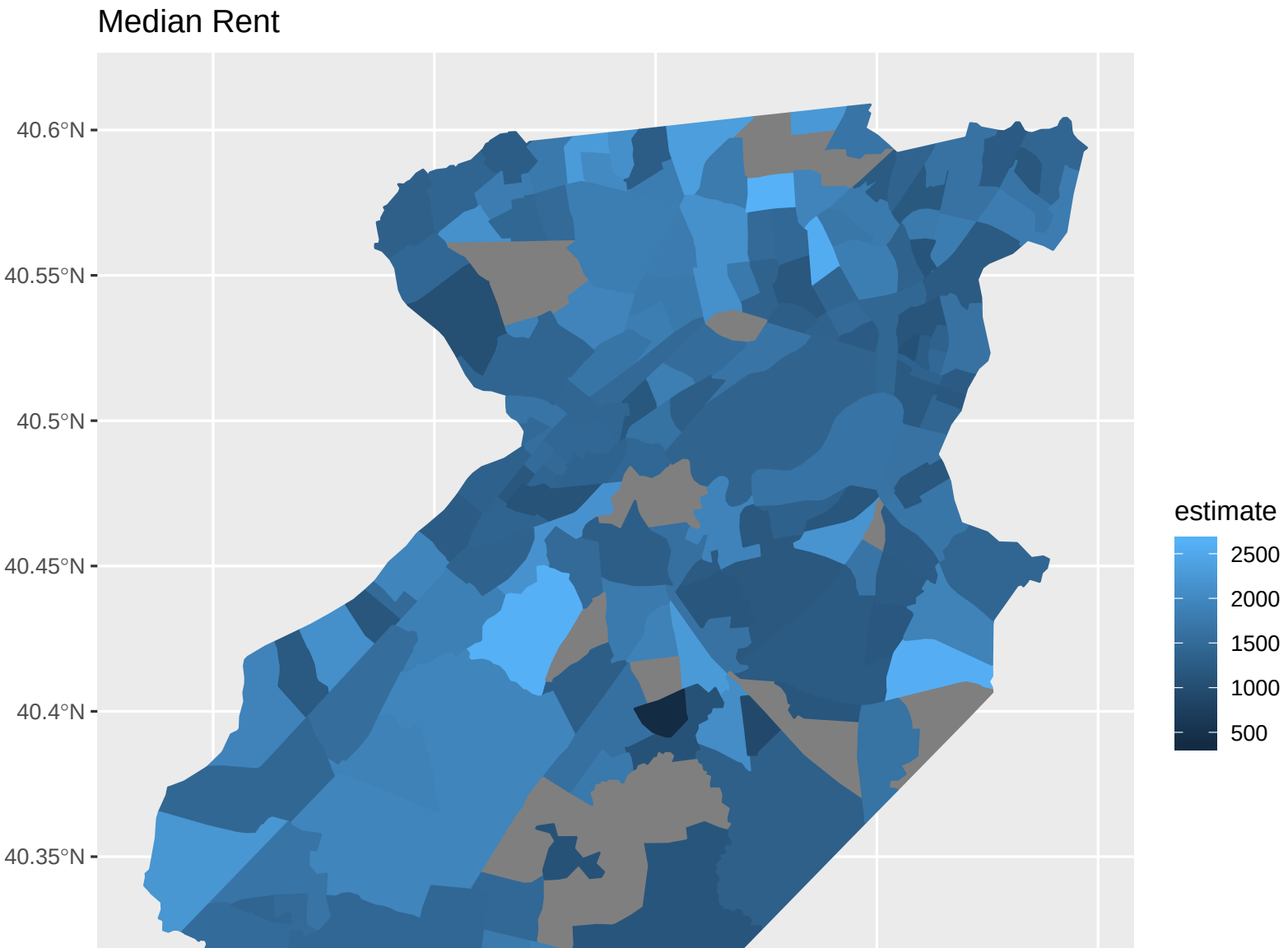
```
## Getting data from the 2014-2018 5-year ACS

## Downloading feature geometry from the Census website.  T
```

```
glimpse(middlesex_rental)
```

```
## Observations: 175
## Variables: 6
## $ GEOID    <chr> "34023000100", "34023000200", "3402300
## $ NAME     <chr> "Census Tract 1, Middlesex County, New
## $ variable <chr> "medrental", "medrental", "medrental",
## $ estimate <dbl> 1321, 1414, 1267, 1792, 2111, 1434, 173
## $ moe      <dbl> 118, 165, 219, 502, 107, 60, 452, 75, N
## $ geometry <MULTIPOLYGON [°]> MULTIPOLYGON (((-74.52667
```

```
middlesex_rental %>%
  ggplot(aes(fill = estimate, color = estimate)) +
  geom_sf() +
  ggtitle("Median Rent")
```



Leaflet

- Leaflet is a very popular set of Javascript libraries for interactive maps.
- There is a nice R package `leaflet`, to interface with them.
- There are lots of details at https://rstudio.github.io/leaflet/

To quote that webpage, the basic usage is

1. Create a map widget by calling leaflet().
2. Add layers (i.e., features) to the map by using layer functions (e.g. `addTiles`, `addMarkers`, `addPolygons`) to modify the map widget.
3. Repeat step 2 as desired.
4. Print the map widget to display it.

```r
pal <- colorQuantile("Greens", domain = nj_median_household

nj_median_household_income2 %>%
    st_transform(crs = "+init=epsg:4326") %>%
    leaflet() %>%
    addProviderTiles(provider = "CartoDB.Positron") %>%
    addPolygons(popup = ~ str_extract(NAME, "^([^,]*)"),
                stroke = FALSE,
                smoothFactor = 0,
                fillOpacity = 0.5,
                color = ~ pal(estimate)) %>%
    addLegend("bottomright",
              pal = pal,
              values = ~ estimate,
              title = "Median Household Income",
              opacity = 1)
```

In class assignment

- Generate a leaflet interactive map of rent (at the census tract level) in NJ.