A Pragmatic Approach to Census Analysis: Tidycensus and R

Jamaal Green

September 19, 2017

Introduction

About Me

- PhD Candidate in Urban Studies and Planning

About Me

- PhD Candidate in Urban Studies and Planning
- My dissertation is examining industrial zoning and labor market change

About Me

- PhD Candidate in Urban Studies and Planning
- My dissertation is examining industrial zoning and labor market change
- I use a pretty wide array of census products for work (ACS, PUMS, LEHD)

A Pragmatic Approach

"Let the question guide your method"

Likewise. . .

Let your problems guide your tools

What's your workflow normally resemble?

The steps many social data analysts and GIS user have to make:

- Tabular Data collection/import (factfinder)

What's your workflow normally resemble?

The steps many social data analysts and GIS user have to make:

- Tabular Data collection/import (factfinder)
- Spatial Data collection(Tigerline files, anyone?)

What's your workflow normally resemble?

The steps many social data analysts and GIS user have to make:

- Tabular Data collection/import (factfinder)
- Spatial Data collection(Tigerline files, anyone?)
- Tabular data cleaning, munging, and joins

What's your workflow normally resemble?

The steps many social data analysts and GIS user have to make:

- Tabular Data collection/import (factfinder)
- Spatial Data collection(Tigerline files, anyone?)
- Tabular data cleaning, munging, and joins
- Table to spatial data joins (we've all done this in Arc with moderate success)

What's your workflow normally resemble?

The steps many social data analysts and GIS user have to make:

- Tabular Data collection/import (factfinder)
- Spatial Data collection(Tigerline files, anyone?)
- Tabular data cleaning, munging, and joins
- Table to spatial data joins (we've all done this in Arc with moderate success)
- If making maps. . . spatial processing (clips, intersections, spatial joins)

What's your workflow normally resemble?

The steps many social data analysts and GIS user have to make:

- Tabular Data collection/import (factfinder)
- Spatial Data collection(Tigerline files, anyone?)
- Tabular data cleaning, munging, and joins
- Table to spatial data joins (we've all done this in Arc with moderate success)
- If making maps. . . spatial processing (clips, intersections, spatial joins)
- Other visualizations and report writing

This Works But It Could be Better...

This workflow is effective but suffers from:

1. Massive number of intermediate outputs

This Works But It Could be Better...

This workflow is effective but suffers from:

1. Massive number of intermediate outputs
2. Jumps among any number of different applications making confusion likely

This Works But It Could be Better...

This workflow is effective but suffers from:

1. Massive number of intermediate outputs
2. Jumps among any number of different applications making confusion likely
3. Can easily become disorganized if data/project management isn't specified beforehand

Enter R

What is R?

- A powerful language

What is R?

- A powerful language
- Application

What is R?

- A powerful language
- Application
- “Do it all” workbench

But why R?

- It's free

But why R?

- It's free
- It's fast

But why R?

- It's free
- It's fast
- It's data type agnostic (read any variety of text files, .shp. GEOJSON, GEOTIFF)

But why R?

- It's free
- It's fast
- It's data type agnostic (read any variety of text files, .shp. GEOJSON, GEOTIFF)
- Massive number of packages for statistical or spatial analysis and visualization

But why R?

- It's free
- It's fast
- It's data type agnostic (read any variety of text files, .shp. GEOJSON, GEOTIFF)
- Massive number of packages for statistical or spatial analysis and visualization
- Many things that are hard or slow in other applications (table joins in Arc, anyone?) are fast in R

But why R?

- It's free
- It's fast
- It's data type agnostic (read any variety of text files, .shp. GEOJSON, GEOTIFF)
- Massive number of packages for statistical or spatial analysis and visualization
- Many things that are hard or slow in other applications (table joins in Arc, anyone?) are fast in R
- Large, helpful online community and growing variety of books/guides/courses

But why R?

- It's free
- It's fast
- It's data type agnostic (read any variety of text files, .shp. GEOJSON, GEOTIFF)
- Massive number of packages for statistical or spatial analysis and visualization
- Many things that are hard or slow in other applications (table joins in Arc, anyone?) are fast in R
- Large, helpful online community and growing variety of books/guides/courses
- IT'S FREE

But why should I?

Has the following ever happened to you?

- Need to download multiple variables over multiple years and you get a data folder filled with ambiguously named tables that you end up deleting anyway?

But why should I?

Has the following ever happened to you?

- Need to download multiple variables over multiple years and you get a data folder filled with ambiguously named tables that you end up deleting anyway?
- Had to change your geography of interest on short notice and then go through the time consuming process of redownloading and processing?

But why should I?

Has the following ever happened to you?

- Need to download multiple variables over multiple years and you get a data folder filled with ambiguously named tables that you end up deleting anyway?
- Had to change your geography of interest on short notice and then go through the time consuming process of redownloading and processing?
- Attempted to rename a column in ArcMap (yes, I know this is now available in ArcPro)?

Let's Be Pragmatic

These recurring challenges can be better addressed (saving yourself precious time) by learning a little bit of R

Enter the tidyverse and tidycensus. . . A Better Way

tidyverse

"An opinionated collection of R packages for data science"

A set of packages to handle common "data science" tasks with consistent behavior and language. A more accessible way to do data science in R for all steps of a project from data import/cleaning to visualization and modeling

Tidycensus. . . one stop shop for ACS data

- R package authored by Prof. Kyle Walker at TCU to make gathering and visualizing census data easier

Tidycensus. . . one stop shop for ACS data

- R package authored by Prof. Kyle Walker at TCU to make gathering and visualizing census data easier
- The package uses the census API to call ACS and decennial data as well as ACS Data Profile tables

Tidycensus. . . one stop shop for ACS data

- R package authored by Prof. Kyle Walker at TCU to make gathering and visualizing census data easier
- The package uses the census API to call ACS and decennial data as well as ACS Data Profile tables
- Data is returned in either wide or long format and there is an option to join the data to its appropriate Tigerline geometry

A quick example. . .

Our assignment: Get latest 5 year MHI for Multnomah County at Tract Level and graph the results (as an added bonus, and in the interest of transparency, let's include MOEs)

Query ACS

```
if(!require(pacman)){install.packages("pacman");
  library(pacman)}
p_load(ggplot2, tidycensus, dplyr)

acs_key <- Sys.getenv("CENSUS_API_KEY")

#Enter the variables and geographies below
census_title <- c("Median Household Income by County:\n
Coefficient of Variation")
census_var <- c("B19013_001E")
census_geog <- c("county")
census_state <- c("or")


acs_data <- get_acs(geography = census_geog, variables =
census_var, state = census_state, output = "wide")
```

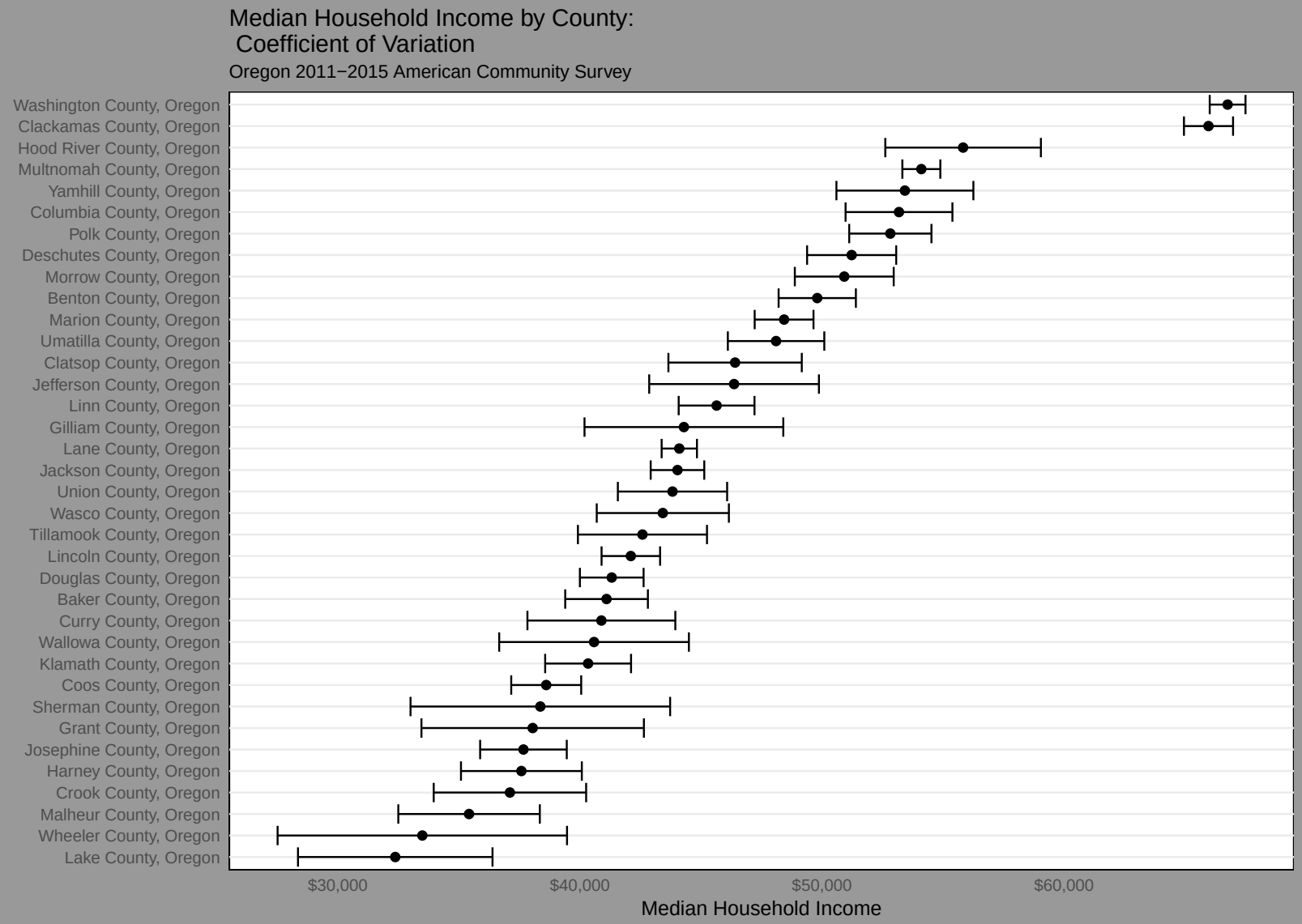
Little Bit of Processing

```
#Make more readable column names
acs_data <- acs_data %>%
  rename(MHI_est = B19013_001E ,
   MHI_moe = B19013_001M)

#Calculate the SE, CV for future reference
acs_data <- acs_data %>%
  mutate(se = MHI_moe/1.645,
         cv = (se/MHI_est)*100)
```

Finally. . . let's plot

```
#Plot Percentages with Derived MOE
acs_plot <- acs_data %>%
  ggplot(aes(x = MHI_est,
  y = reorder(NAME, MHI_est))) +
  geom_point(color = "black", size = 2) +
  geom_errorbarh(aes(xmin = MHI_est - MHI_moe,
                     xmax = MHI_est + MHI_moe )) +
  labs(title = paste(census_title),
       subtitle =
      paste0("Oregon 2011-2015 American Community Survey"),
       x = "Median Household Income") +
  scale_x_continuous(labels = scales::dollar) +
  theme_minimal() +
  theme(panel.grid.minor.x  = element_blank(),
        panel.grid.major.x = element_blank())


plot(acs_plot)
```

Our Output


Median Household Income by County:
Coefficient of Variation
Oregon 2011−2015 American Community Survey
Washington County, Oregon
Clackamas County, Oregon
Hood River County, Oregon
Multnomah County, Oregon
Yamhill County, Oregon
Columbia County, Oregon
Polk County, Oregon
Deschutes County, Oregon
Morrow County, Oregon
Benton County, Oregon
Marion County, Oregon
Umatilla County, Oregon
Clatsop County, Oregon
Jefferson County, Oregon
Linn County, Oregon
Gilliam County, Oregon
Lane County, Oregon
Jackson County, Oregon
Union County, Oregon
Wasco County, Oregon
Tillamook County, Oregon
Lincoln County, Oregon
Douglas County, Oregon
Baker County, Oregon
Curry County, Oregon
Wallowa County, Oregon
Klamath County, Oregon
Coos County, Oregon
Sherman County, Oregon
Grant County, Oregon
Josephine County, Oregon
Harney County, Oregon
Crook County, Oregon
Malheur County, Oregon
Wheeler County, Oregon
Lake County, Oregon
$30,000
$40,000
$50,000
$60,000
Median Household Income

Mapping It Out

R as a GIS- tigris and sf

tigris- a package that will download tigerline shapefiles

simple features- uses well known text to signify geometry allowing for spatial objects to be treated as dataframes

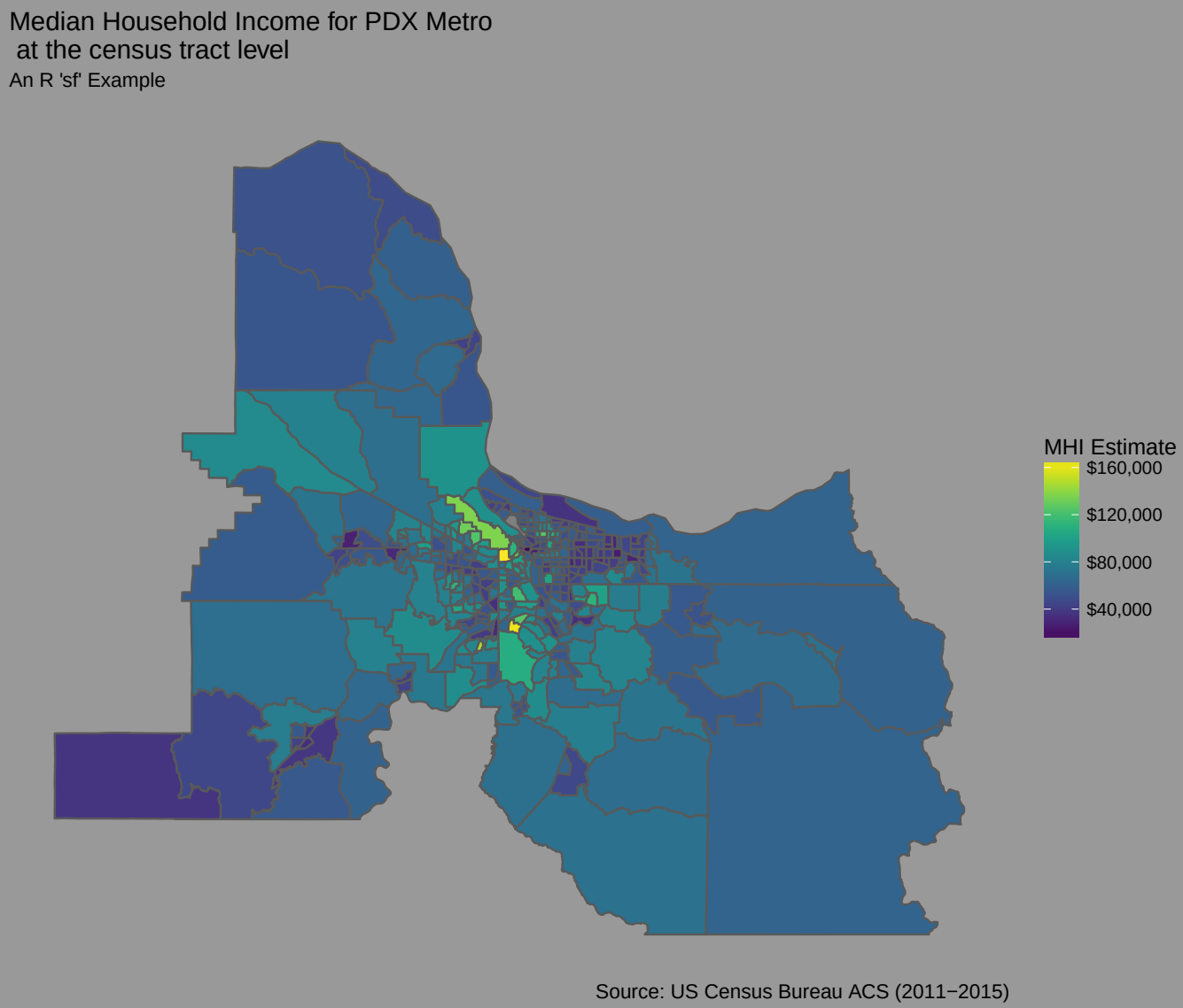
Tract Processing tidyverse style

```
if(!require(pacman)){install.packages("pacman");
  library(pacman)}
p_load(sf, tigris, viridis, ggthemes, ggplot2,
       tidycensus, stringr, dplyr)
options(tigris_class = "sf", tigris_use_cache = TRUE)

acs_key <- Sys.getenv("CENSUS_API_KEY")

mhi_tables <- c("B19013_001")

#download tracts and county, get the tracts for PDX
#Metro counties and counties for the state

mhi_tract <- get_acs(geography = "tract",
                     variables = mhi_tables,
                     state = "OR",
                     geometry = TRUE)
```

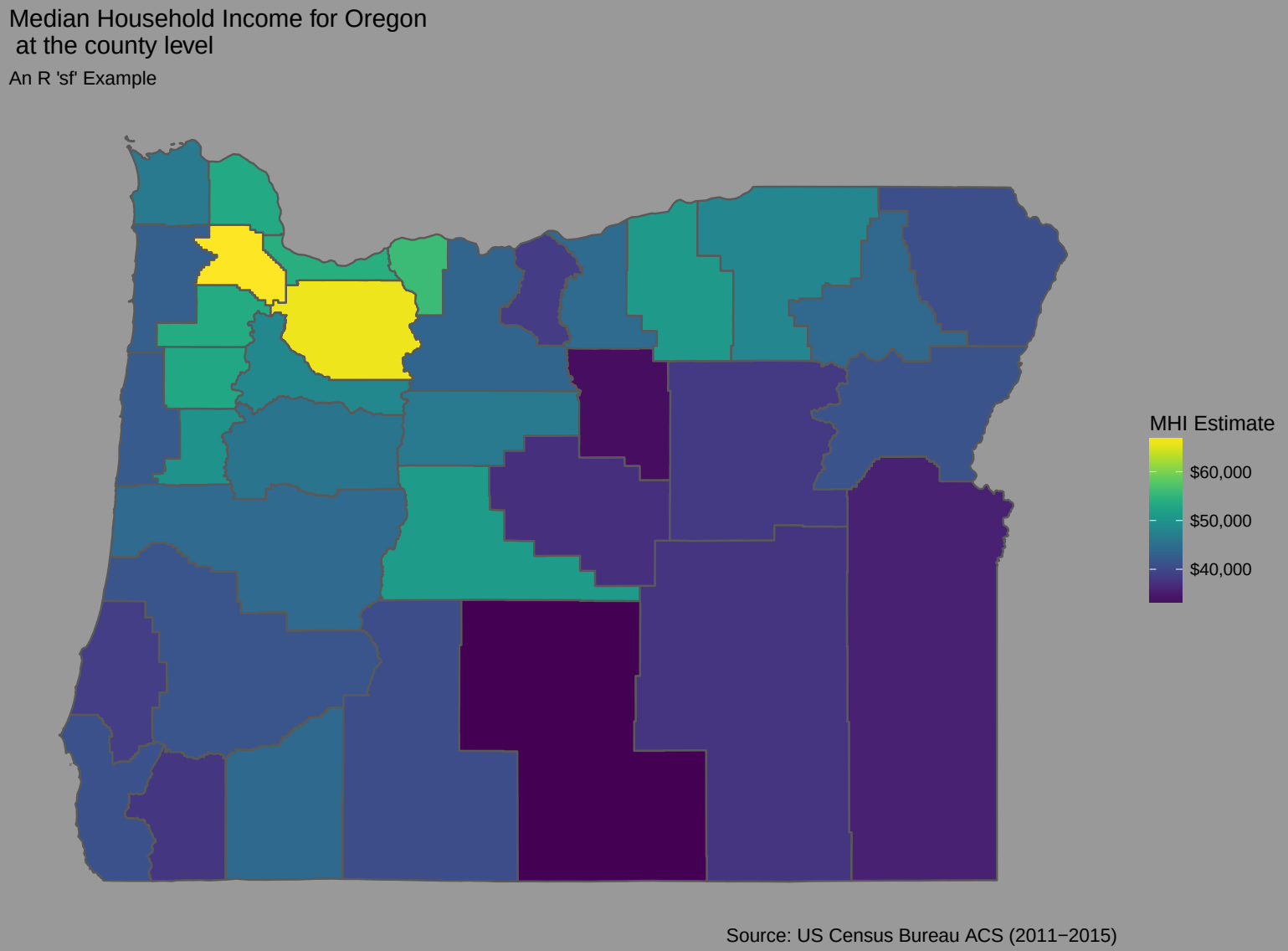
Our Tract Map Set Up

```
p1 <- ggplot() +
  geom_sf(data = metro_tract, aes(fill = estimate)) +
  coord_sf(datum = NA) +
  theme(plot.title = element_text(size = 16,
                    face = "bold", margin = margin(b=10))) +
  theme(plot.subtitle = element_text(size = 14,
                      margin = margin(b = -20))) +
  theme(plot.caption = element_text(size = 9,
                    margin = margin(t = -15), hjust = 0)) +
  scale_fill_viridis(labels = scales::dollar,
                     name = "MHI Estimate") +
  labs(caption = "Source: US Census Bureau ACS (2011-2015)",
       title = "Median Household Income for PDX Metro\n at th
       subtitle = "An R 'sf' Example") + theme_minimal()
```

Our Tract Output



Our County Output



Let's Stretch A Bit

Your Assignment

Create a social vulnerability index for the PDX Metro area using % in poverty, non-White %, % under 5 years of age, and % over 64 years of age and map it

Inital Collection and Processing

```
#name the tables we need
vul_vars <- c("B17001_001", "B17001_002",  "B02001_001",
              "B02001_002", "B01001_003", "B01001_020",
              "B01001_021", "B01001_022", "B01001_023",
              "B01001_024", "B01001_025","B01001_027",
              "B01001_044", "B01001_045", "B01001_046",
              "B01001_047", "B01001_048", "B01001_049")

#grab the data for Oregon

vul_acs <-
  get_acs(geography = "tract", variables = vul_vars,
  state = "OR", output = "wide")

vul_acs <- vul_acs %>%
  mutate(CountyFIPS = str_sub(GEOID, 1, 5))
```

Clean Up Table and Calculate Percentages

```
vul2 <- vul_acs %>%
  mutate(PovShare = B17001_002E/B17001_001E,
         NonWhite = (B02001_001E - B02001_002E)/B02001_001E,
         Under5 = (B01001_003E + B01001_027E)/B02001_001E,
         Older64Male = B01001_020E + B01001_021E +
         B01001_022E +B01001_023E + B01001_024E +
         B01001_025E,
         Older64Female = B01001_044E +
         B01001_045E + B01001_046E + B01001_047E +
         B01001_048E + B01001_049E,
         Older64 =
         (Older64Male + Older64Female)/B02001_001E) %>%
  select(NAME, GEOID, CountyFIPS, PovShare,
          NonWhite, Under5, Older64)
```

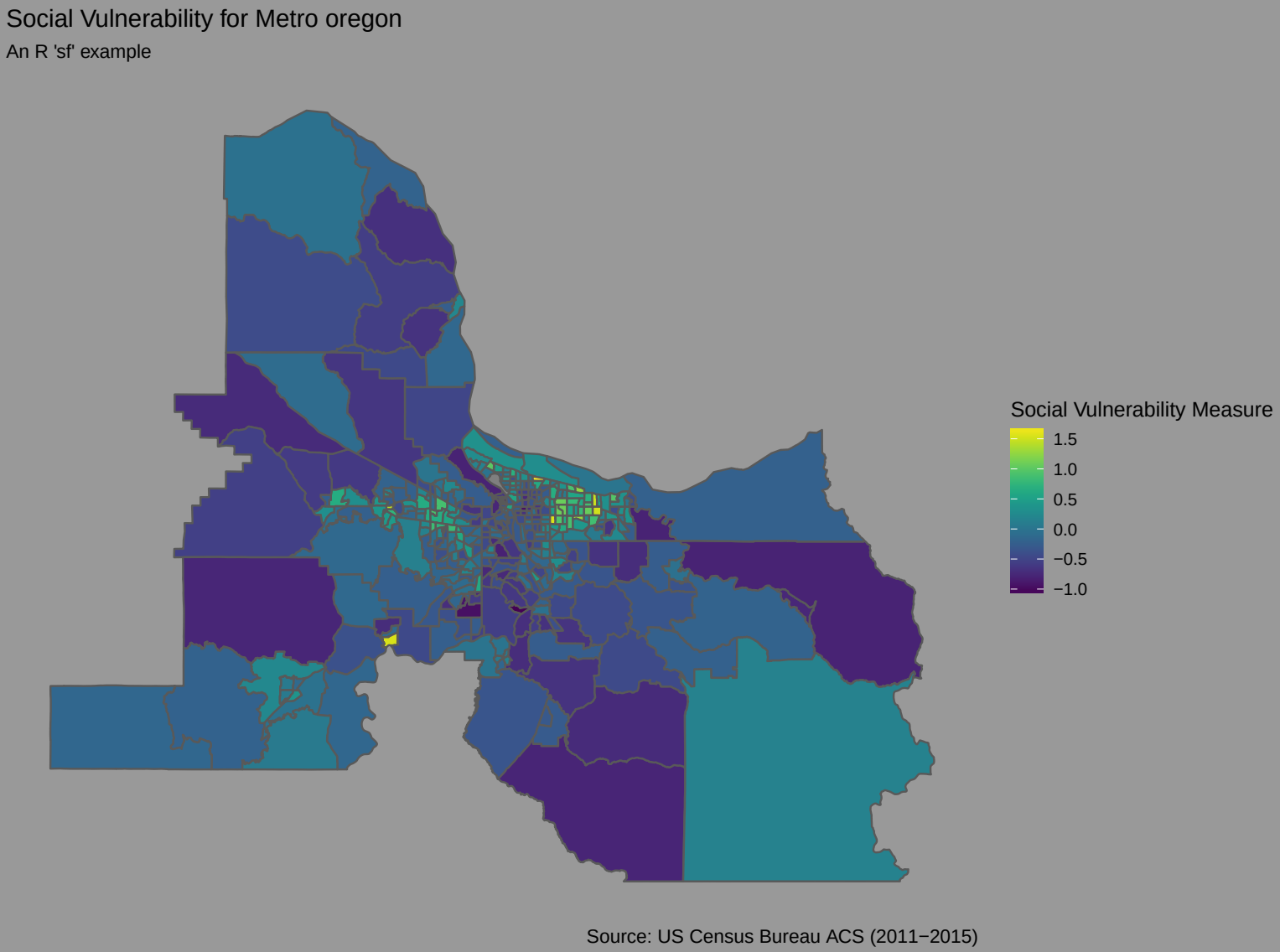
Get Index Values

```
vul2 <- vul2 %>%
  mutate(
  z_Pov = (PovShare - mean(PovShare, na.rm = TRUE))/
            sd(PovShare, na.rm = TRUE),
  z_NonWhite = (NonWhite - mean(NonWhite, na.rm = TRUE))/
            sd(NonWhite, na.rm = TRUE),
  z_Under5 = (Under5 - mean(Under5, na.rm = TRUE))/
            sd(Under5, na.rm = TRUE),
  z_Older64 = (Older64 - mean(Older64, na.rm = TRUE))/
            sd(Older64, na.rm = TRUE))

vul2 <- vul2 %>%
  mutate(VulIndex = (z_Pov + z_NonWhite + z_Under5 +
                       z_Older64)/4) %>%
  select(GEOID, CountyFIPS, z_Pov, z_NonWhite, z_Under5,
         z_Older64, VulIndex)
```

Subset & Join to Geometry

```
metro_counties <- c("41051", "41005", "41009",
                    "41067", "41071")

vul2 <- vul2 %>%
  filter(CountyFIPS %in% metro_counties)

or_tracts <- tracts(state = "OR")

metro_vul <- inner_join(vul2, or_tracts,
                        by = c("GEOID" = "GEOID")) %>%
  select(1:7, geometry) %>% st_as_sf()
```

And. . . voila


Social Vulnerability for Metro oregon
An R 'sf' example
Social Vulnerability Measure
1.5
1.0
0.5
0.0
−0.5
−1.0
Source: US Census Bureau ACS (2011−2015)

Wrapping Up

Let the Problem Guide The Tool

It's not necessary to do everything in R, but we can do a lot of things faster and easier in R

A tidy approach. . .

The tidyverse, and packages connected to it, make R more approachable than ever

Some Additional Resources

- "R For Data Science" by Wickham and Grolemund
- DataCamp
- StackOverflow
- Presentation Link