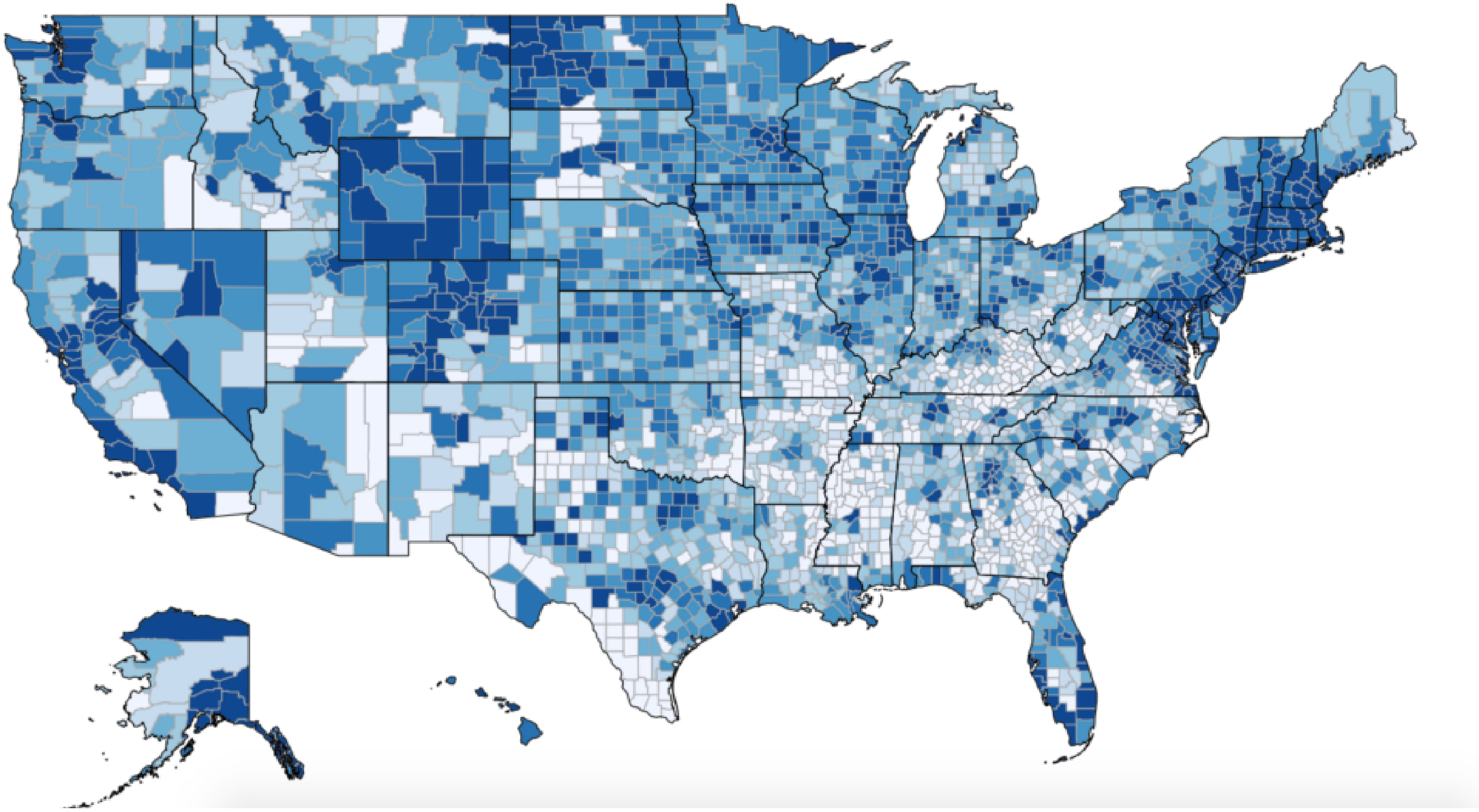
3 Simple Steps to Map Geospatial Data in R

1) Get API 2) Install Libraries 3) Graph

Amanda West Follow
Jun 22 · 3 min read ★



I always wondered how people created beautiful geospatial visualizations in R. I would stare googly-eyed at the perfect gradients which could make a dashboard or presentation pop, wondering just *how the hell* people were doing this in the programming language I associated with p-values and regression lines.

Surprisingly, you can actually accomplish this with but a few lines of code — I'm here today to show you how.

Step 1: Access the American Community Survey ("acs") API

First, you'll need access to the Census Bureau's API. Go to this link to request an API key. After about 10 seconds you should get an email with your key and a link to confirm you are, indeed, human. Stash your API key somewhere — you'll need to call it every time you reopen R.

In order for the API to be read correctly, we'll need to install 3 libraries: "acs", "choroplethr", and "choroplethrMaps". We'll also install "RColorBrewer" and "ggplot2" for graphing later on.

> *A **choropleth** map (from Greek χῶρος "area/region" and πλῆθος "multitude") is a type of thematic map in which areas are shaded or patterned in proportion to a statistical variable that represents an aggregate summary of a geographic characteristic within each area, such as population density or per-capita income.*— Wikipedia

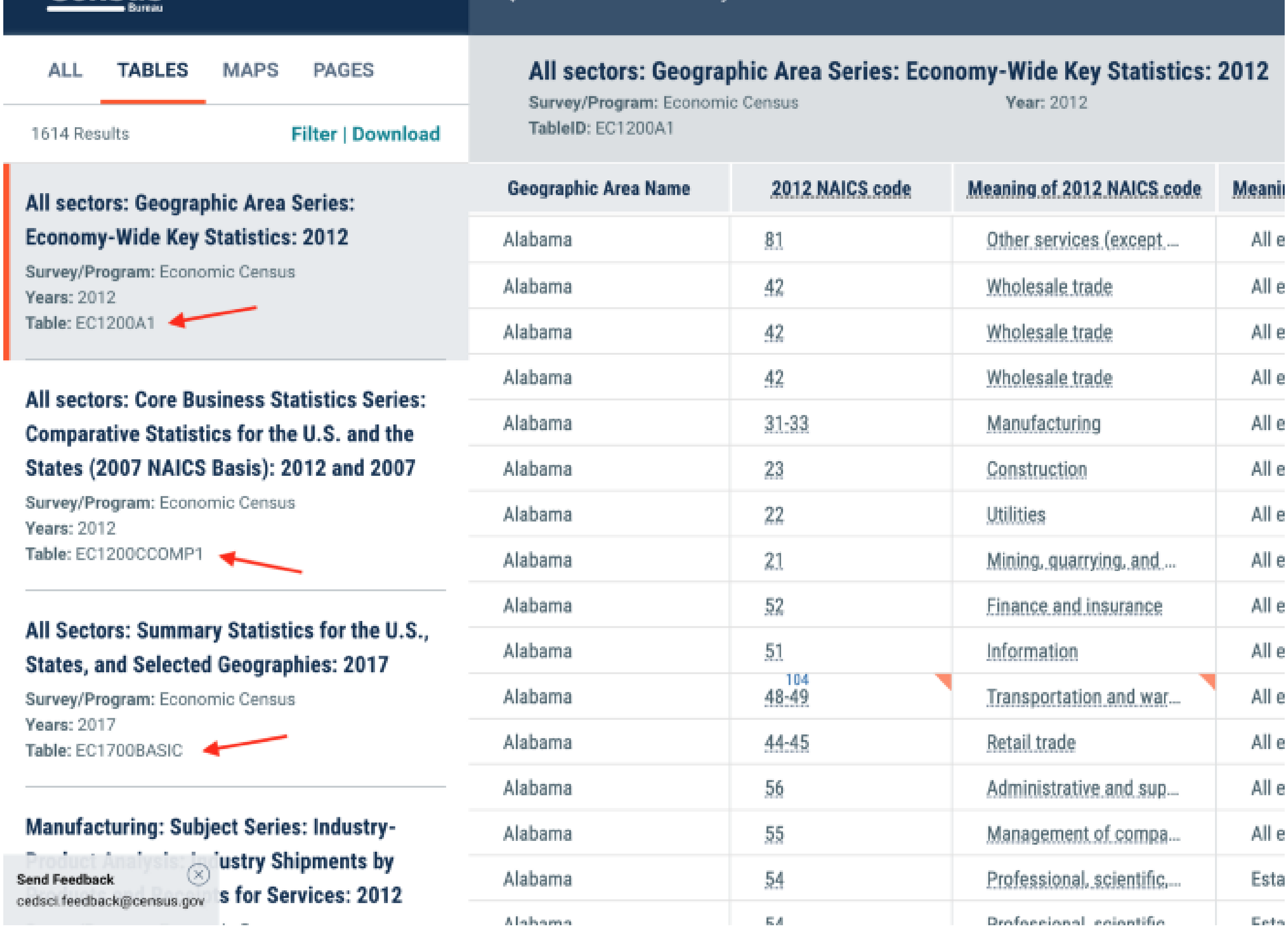
Here is the code you'll need to input into R to get the API to work:

```
# Geospatial Visualization in R

install.packages(c("acs",
                   "choroplethr",
                   "choroplethrMaps",
                   "RColorBrewer",
                   "ggplot2"))

library(acs)
library(choroplethr)
library(choroplethrMaps)
library(RColorBrewer)
library(ggplot2)

api.key.install("<insert personal API key here>")
```

Geospatial R Installation hosted with ♡ by GitHub view raw

Step 2: Choose table ID

Once you have the API key installed, you can find available datasets at data.census.gov to play around with (I am using table B19301, per capita income by county, for the tutorial). When you find data you want to use, take note of the table ID. The table ID is all you will need to create a beautiful choropleth graph.



1614 Results

Filter | Download

All sectors: Geographic Area Series: Economy-Wide Key Statistics: 2012
Survey/Program: Economic Census
Years: 2012
Table: EC1200A1

All sectors: Core Business Statistics Series: Comparative Statistics for the U.S. and the States (2007 NAICS Basis): 2012 and 2007
Survey/Program: Economic Census
Years: 2012
Table: EC1200CCOMP1

All Sectors: Summary Statistics for the U.S., States, and Selected Geographies: 2017
Survey/Program: Economic Census
Years: 2017
Table: EC1700BASIC

Manufacturing: Subject Series: Industry-Product Analysis: Industry Shipments by
s for Services: 2012

Send Feedback
cedsci.feedback@census.gov

All sectors: Geographic Area Series: Economy-Wide Key Statistics: 2012
Survey/Program: Economic Census Year: 2012
TableID: EC1200A1

Geographic Area Name	2012 NAICS code	Meaning of 2012 NAICS code	Meani
Alabama	81	Other services (except...	All e
Alabama	42	Wholesale trade	All e
Alabama	42	Wholesale trade	All e
Alabama	42	Wholesale trade	All e
Alabama	31-33	Manufacturing	All e
Alabama	23	Construction	All e
Alabama	22	Utilities	All e
Alabama	21	Mining, quarrying, and ...	All e
Alabama	52	Finance and insurance	All e
Alabama	51	Information	All e
Alabama	104 48-49	Transportation and war...	All e
Alabama	44-45	Retail trade	All e
Alabama	56	Administrative and sup...	All e
Alabama	55	Management of compa...	All e
Alabama	54	Professional, scientific,...	Esta
Alabama	54	Professional, scientific	Esta

After you've settled on a dataset, let's get graphing!

Step 3: Mapping Geospatial Data

Below I have the code to graph U.S. income per capita by county. This is only a glimpse of what can be toggled; if you have a modification I didn't exemplify, the R documentation for county_choropleth_acs and ggplot2 Colors will get you the rest of the way there.

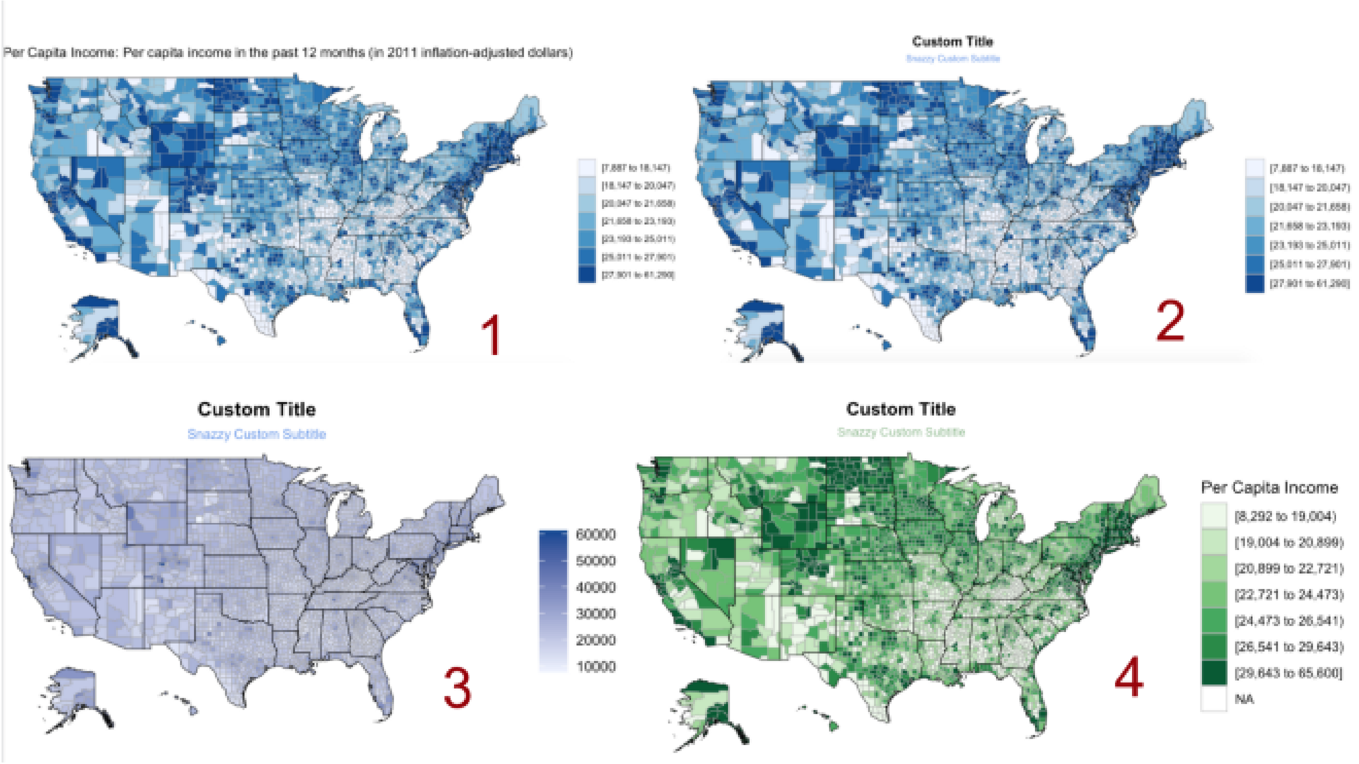
- Some ACS datasets contain null values; the most colorful graphs will be those with most of their observations present!

```
# 1 || Basic / Default
county_choropleth_acs(tableId = "B19301")

# 2 || Add a custom title + subtitle
county_choropleth_acs(tableId = "B19301") +
  labs(title="Custom Title",
       subtitle = "Snazzy Custom Subtitle") +
  theme(
       plot.title = element_text(size = 12, face = "bold", hjust = 0.5),
       plot.subtitle = element_text(size = 8, color = "cornflowerblue", hjust = 0.5))

```

```
# 3 || Continuous Gradient
county_choropleth_acs(tableId = "B19301", num_colors = 1) +
  labs(title="Custom Title",
       subtitle = "Snazzy Custom Subtitle") +
  theme(
    plot.title = element_text(size = 12, face = "bold", hjust = 0.5),
    plot.subtitle = element_text(size = 8, color = "cornflowerblue", hjust = 0.5))

# 4 || Change Color to Green
col.pal<-brewer.pal(7,"Greens")
df_2015<-get_acs_data(tableId="B19301",map="county",endyear=2015)
choro1<-CountyChoropleth$new(df_2015$df)
choro1$ggplot_scale <- scale_fill_manual(name="Per Capita Income",values=col.pal, drop=FALSE)
choro1$render() +
  labs(title="Custom Title",
       subtitle = "Snazzy Custom Subtitle") +
  theme(
    plot.title = element_text(size = 12, face = "bold", hjust = 0.5),
    plot.subtitle = element_text(size = 8, color = "darkseagreen", hjust = 0.5))
```

Graphing Choropleths hosted with ♡ by GitHub view raw

….and the related graphs:



\\ 1: Default Values \\ 2: Custom Title \\ 3: Gradient + Custom Title \\ 4: Color Change + Custom Title \\

Bonus: States & Regions Only

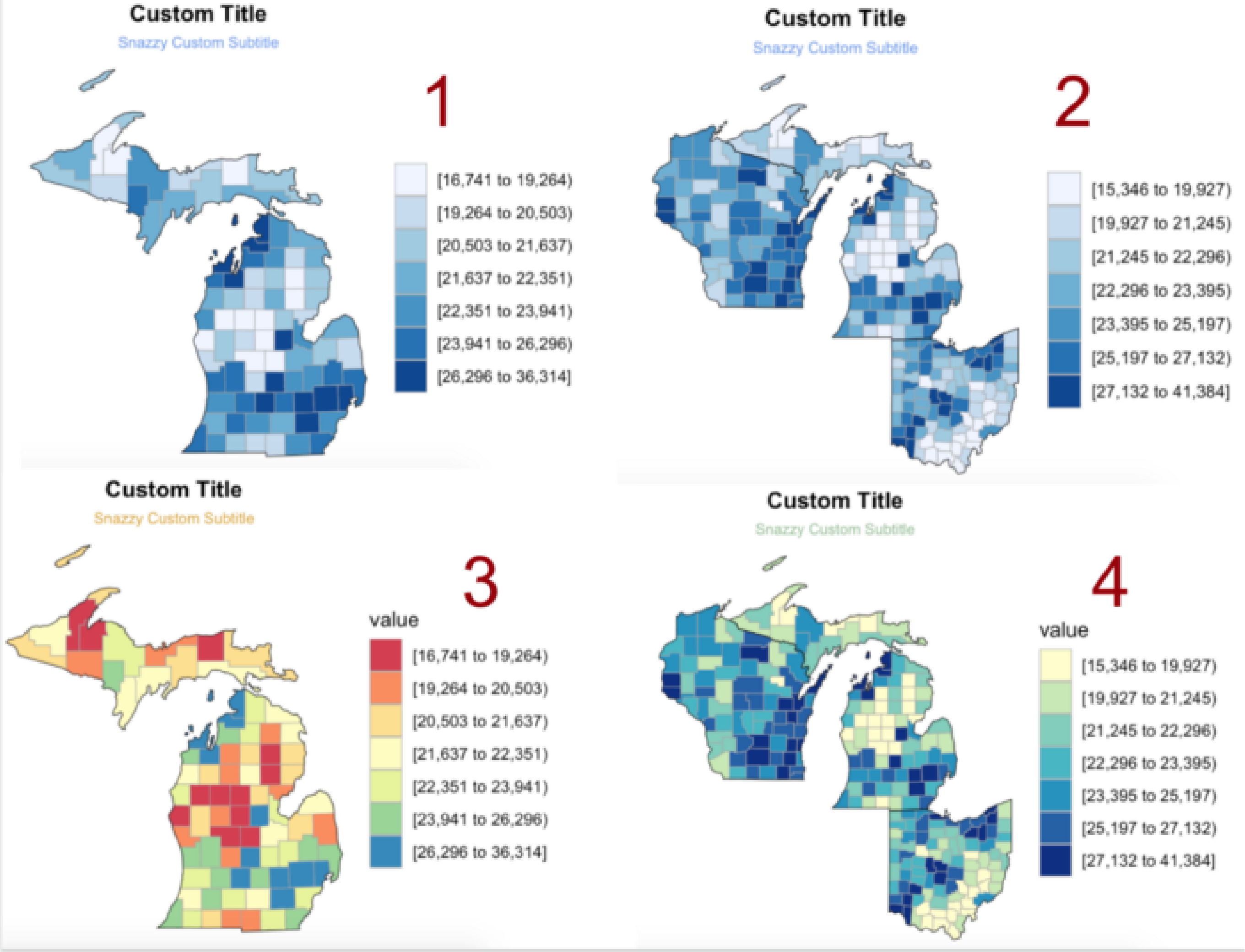
Changing our graphs to focus on a state or region is as easy as adding one additional input, the `state_zoom()` function. In order to graph more than one state at once (but not the whole country), we'll just concatenate all the states we want to view together.

For the last two graphs, I used some fun palettes provided by ggplot2. This library's `scale_fill_brewer()` function contains a set of predetermined palettes which you can find here.

```
# 1 || Specify Michigan
county_choropleth_acs(tableId = "B19301", state_zoom="michigan") +
  labs(title="Custom Title",
       subtitle = "Snazzy Custom Subtitle") +
  theme(
    plot.title = element_text(size = 12, face = "bold", hjust = 0.5),
    plot.subtitle = element_text(size = 8, color = "cornflowerblue", hjust = 0.5))

# 2 || Specify a Mix of States
county_choropleth_acs(tableId = "B19301", state_zoom=c("michigan", "ohio", "wisconsin")) +
  labs(title="Custom Title",
       subtitle = "Snazzy Custom Subtitle") +
  theme(
    plot.title = element_text(size = 12, face = "bold", hjust = 0.5),
    plot.subtitle = element_text(size = 8, color = "cornflowerblue", hjust = 0.5))

# 3 || Change Michigan Colors with scale_fill_brewer()
county_choropleth_acs(tableId = "B19301", state_zoom=c("michigan")) +
  scale_fill_brewer(palette="Spectral") +
  labs(title="Custom Title",
       subtitle = "Snazzy Custom Subtitle") +
  theme(
    plot.title = element_text(size = 12, face = "bold", hjust = 0.5),
    plot.subtitle = element_text(size = 8, color = "goldenrod", hjust = 0.5))

# 4 || Change State Group Colors with scale_fill_brewer()
county_choropleth_acs(tableId = "B19301", state_zoom=c("michigan", "ohio", "wisconsin")) +
  scale_fill_brewer(palette="YlGnBu") +
  labs(title="Custom Title",
       subtitle = "Snazzy Custom Subtitle") +
  theme(
    plot.title = element_text(size = 12, face = "bold", hjust = 0.5),
    plot.subtitle = element_text(size = 8, color = "darkseagreen", hjust = 0.5))
```

Choropleths by State, Region hosted with ♡ by GitHub view raw

….and here are the four related graphs:



Conclusions

Now you know how to create simple geospatial visualizations in R. You can install the Census API and pick a dataset (from literally hundreds!) you'd like to investigate. You can graph by country, state, or a smattering of states. Finally, you can change the palette, gradient, and titles of your graphs.

This is only the tip of the geospatial iceberg, but I hope it can get you started. Thanks for reading and have a great work week! ♡

GIF from GIPHY

Data Science Programming Geospatial Visualization Rstudio