### **Exercise 2. Data Management Fundamentals**

Part 2. Independently Accessing an API and Creating your Own Plots

#### Summary

The purpose of lab 2 is to introduce you to the concept of an application program interface (API) as well as to manage (& tidy up) the data you obtain. First you learned how to retrieve public data from a remote server provided by the Federal Census Bureau using R and an API. Then you bound your data together, convert it from wide to long format and used ggplot to generate a plot. Now you will go back to a different API, retrieve the data and again use ggplot to generate your own plot.

#### Objectives

- 1. To interactively retrieve data from a remote server using an application program interface (API) and R.
- 2. To use the tidyr package to convert multivariable datasets from wide to long format.
- 3. To cut the dataset into quantiles and plot the results as facets of the dataset.

#### Grading

This is part two of lab 2 and is due on Saturday, February 23rd by 10:00 PM. This lab is worth 10 points towards your final grade. You will need to provide the deliverable from part 1, as well as deliverables 2 & 3 (found below). In addition to these 3 deliverables and the criteria specified and enumerated within the laboratory instructions, your grade will be based on the following criteria.

- Your ability to obtain data from a remote server using an application program interface.
- Your ability to plot, describe and communicate the datasets as requested.
- Your ability to manipulate and transform the datasets in order to produce intended results.
- Your ability to conduct a basic quantitative analysis of the population data in order to cut the dataset into facets of quantiles.

You will need to upload the laboratory deliverables to blackboard prior to the deadline.

Instructions. Independently Accessing an API and Creating your Own Plot

- 1. Open R Studio. Create a new .R script, name it script\_2b or something similar and save it to your H:\\lab\_2 folder. Set your working directory.
- 2. With your browser go to the United States Census Bureau, American Community Survey 5-Year Data API. You should be able to simply click on the following link.

https://www.census.gov/data/developers/data-sets/acs-5year.html



DATA 100 / COLL 100

### wieked problems

View the **Detail Tables** links for each survey year. For example, find the available units of analysis (i.e. geographies), variable list, and examples of how to call individual variables. See the following link to learn more about all of the subjects available in the American Community Survey.

### **ACS Subjects**

https://www.census.gov/programs-surveys/acs/guidance/subjects.
html

3. On the American Community Survey 5-Year Data (2009-2015) page, you will find a description of the data that is available through this API for the survey years 2015 through 2009.

Add the following objects to your script.

- A host with the base address api.census.gov/data/
- A series of objects, each one indicating a different survey year. You will only call ACS years 2014 through 2010.
- An object that describes the acs/acs5? dataset.
- An object that indicates you will retrieve the variables state name, NAME and state population, B01001\_001E from the ACS API.
- An object that indicates you will retrieve all states.
- Finally, an object the contains your key.

Use the <code>paste()</code> command to create paths that will be used to retrieve data for each survey year from 2014 to 2010. Then use the from <code>fromJSON()</code> command to retrieve the data and create each of the 5 data frames.

- 4. For each object you just created, assign the first row within the data frame as the name each each column. Then remove the first row from the data frame and reorganize your columns so they ordered state name, then the FIPs code for the state followed by state population for that survey year last.
- 5. Now use the cbind.data.frame() command to bind population totals from each survey year into a single data frame. Your resulting object should have 52 observations and 7 variables.
- 6. For part 2 of this lab you will generate two plots that will be your deliverables for this second half of the homework assignment. For the first plot you will generate quantiles that segment the ranks of total population growth from 2010 until 2014 for each primary U.S. subdivision. To do this, create a new variable in your data frame that subtracts the total population of each state in 2014 from the total population of each state in 2010. Then use the within() command to create a second new variable that indicates the population



DATA 100 / COLL 100

### wieked problems

quantile for each state. Use the  $cut_number()$  command to segment your 52 row data frame into 9 nearly equal facets.

7. For the second plot, you will calculate the annual growth rate for each state and then determine the overall mean of the four year period (again for each state). To calculate the average growth rate for each state, divide the population for each year by the population from the previous year and then subtract 1. Then use the rowMeans() command to calculate the average annual growth rate during the 4 yearly intervals. Use your assignment operator to create a new variable in your data frame that reflects the average annual growth rate of each U.S. state from 2010 until 2014.

```
data_frame$new_variable <- rowMeans(data_frame[,col:col] /
data_frame[,col:col] - 1)</pre>
```

Then again use the within() command to create a fourth new variable that indicates the average annual growth rate quantile for each state. Again, use the cut\_number() command to segment your 52 row data frame into 9 nearly equal facets. Your data frame should now have 11 variables.

8. In order to plot the data using ggplot, you will need to convert your object from this wide format to the long format. Transform your data using the gather() command from the tidyr:: package.

data\_frame\_long <- gather(data\_frame, "year", "population", col\_num:col\_num)</pre>

9. After converting your data into the long format, use ggplot to create your base plotting object.

10. Now plot your object that facets the absolute population growth of all 52 states into 9 nearly equal sets, with each of the nine plots ordered from lowest to highest (or visa-versa if you prefer). Add points, lines, and labels to your plot as well as remove the legend. Use the + facet\_wrap() command to identify the variable you created that describes the 9 different quantiles of absolute population growth from 2010 and 2014 for each state. Set the ncol = argument equal to 3, in order to arrange your 9 plots in a 3 by 3 grid. Don't forget to include scales = "free" argument.

```
g + geom_point() + geom_line() +
theme(legend.position = "none") +
geom_text(size = 2, vjust = -1) +
facet_wrap(~ quantile, ncol = 2)
```

DATA 100 / COLL 100



11. Now plot your object that facets the overall average annual growth for the years 2010 to 2014 of all 52 states, again into 9 nearly equal sets. Ordered from lowest to highest (or visa-versa if you prefer). Add points, lines, and labels to your plot as well as remove the legend. Use the + facet\_wrap() command, set the ncol = 3 and include the scales = "free" argument.

#### Deliverable 2

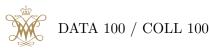
Generate a line plot that facets the absolute population increase between 2010 and 2014 of all 52 primary US subdivisions into 9 quantiles. (see page 5)

#### Deliverable 3

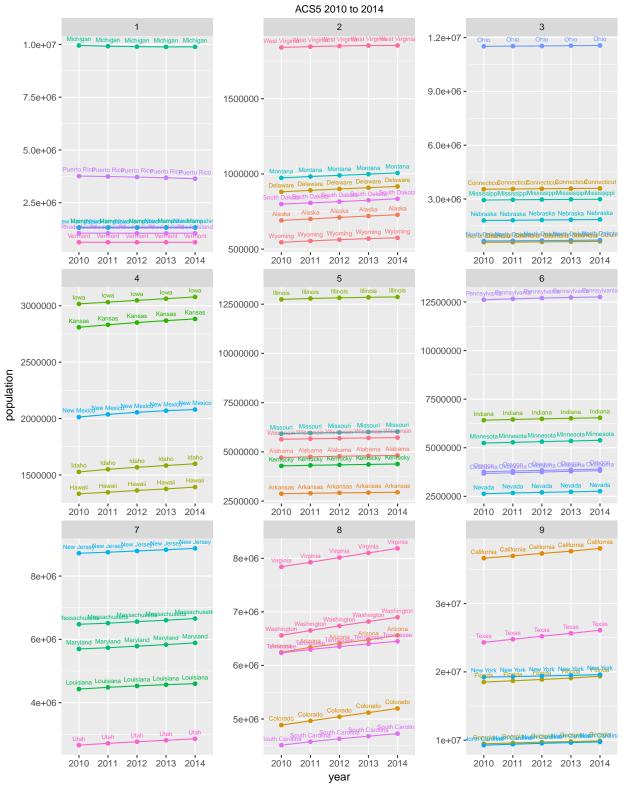
Generate a line plot that facets the average annual growth rate between 2010 and 2014 of all 52 primary US subdivisions into 9 quantiles. (see page 6)

#### Stretch Goal

Go back to the link for the ACS Subject page found on Page 2, and select another variable. Conduct the same analysis at the state level. If you want a real challenge, then also conduct that analysis with your new variable at the county level to describe all 3000+ secondary subdivision units that describe the United States!



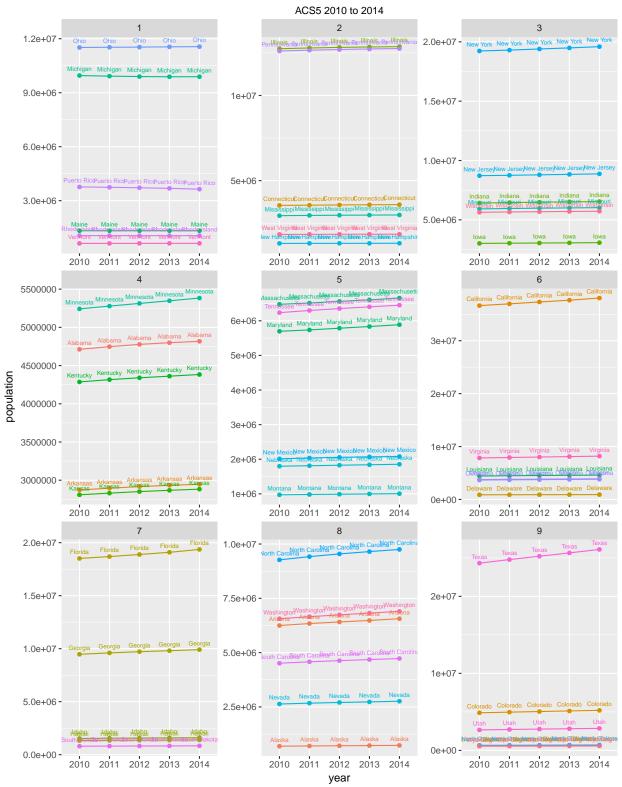
# włoked problems



Absolute Population Increase



# wreked problems



#### Average Annual Growth Rate

