Food Environment Atlas

Put your name here

Put due date here

The Food Environment Atlas is a wide-ranging dataset collected and maintained by the United States Department of Agriculture. Some of the statistics collected give information about health, food insecurity, and food prices and taxes. Data in the Atlas is collected so that researchers at the USDA can better understand national food environments. [More information](https://www.ers.usda.gov/data-products/food-environment-atlas/about-the-atlas/) about the Atlas and detailed [documentation](https://www.ers.usda.gov/data-products/food-environment-atlas/documentation/) are both available.

## Read in the data

The most current version of the Food Environment Atlas is available for download as an excel file from [here](https://www.ers.usda.gov/data-products/food-environment-atlas/data-access-and-documentation-downloads/). The excel file "Data Download" can be downloaded from the link under the "Current Version" header.

The excel file has data stored in several tabs. We can use the readxl package (which is part of the tidyverse) to read in each tab of the downloaded file as separate elements of a list. Once you have installed the packages readxl, rlang, and purrr, you can use the following code to save the "Supplemental Data - County" data frame, which contains county population data, as well as the Insecurity, Prices and Taxes, and Health data frames. Make sure to change the path to where you have the "DataDownload.xls" file saved (in the example code below, we've left the file in the "Downloads" folder). We'll clean up these data frames below to make them easier to use for plotting and analysis. Note that the data is initially in a list format when we read it in from the Excel sheet, so we've put some code in the example code below (class and names) to check the data out when we read it in.

We will also create a data frame with state names and abbreviations called state\_names using a few of R's built-in datasets. state.name is a character vector giving full state names, and state.abb is a character vector of 2-letter abbreviations for the state names. Since states in the Atlas are listed by abbreviation, this data frame will be useful for later cleaning.

path <- "~/Downloads/DataDownload.xls"  
  
food\_list <- path %>%  
 readxl::excel\_sheets() %>%  
 rlang::set\_names() %>%  
 purrr::map(read\_xls, path = path)  
  
class(food\_list)

## [1] "list"

names(food\_list)

## [1] "Read\_Me" "Variable List"   
## [3] "Supplemental Data - County" "Supplemental Data - State"   
## [5] "ACCESS" "STORES"   
## [7] "RESTAURANTS" "ASSISTANCE"   
## [9] "INSECURITY" "PRICES\_TAXES"   
## [11] "LOCAL" "HEALTH"   
## [13] "SOCIOECONOMIC"

ins\_df <- food\_list$INSECURITY  
price\_df <- food\_list$PRICES\_TAXES  
health\_df <- food\_list$HEALTH   
pop\_df <- food\_list$`Supplemental Data - County`  
  
state\_names <- data.frame(state\_name = state.name, state\_abb = state.abb)

# Food insecurity

To begin, clean the State Food Insecurity (ins\_df) data frame. The raw data looks like this:

## # A tibble: 3 x 11  
## FIPS State County FOODINSEC\_10\_12 FOODINSEC\_13\_15 CH\_FOODINSEC\_12\_15  
## <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 01001 AL Autauga 17.9 17.6 -0.3  
## 2 01003 AL Baldwin 17.9 17.6 -0.3  
## 3 01005 AL Barbour 17.9 17.6 -0.3  
## # ... with 5 more variables: VLFOODSEC\_10\_12 <dbl>, VLFOODSEC\_13\_15 <dbl>,  
## # CH\_VLFOODSEC\_12\_15 <dbl>, FOODINSEC\_CHILD\_01\_07 <dbl>,  
## # FOODINSEC\_CHILD\_03\_11 <dbl>

Select and rename the State column and FOODINSC\_10\_12 column, which gives the percent of households experiencing food insecurity in a three-year average from 2010 through 2012. Since the food insecurity data is reported by state, make sure that there is one row per state. (*Hint*: The distinct() function from dplyr is useful for this step.)

## # A tibble: 3 x 2  
## state\_abb percent  
## <chr> <dbl>  
## 1 AL 17.9  
## 2 AK 12.1  
## 3 AZ 19.7

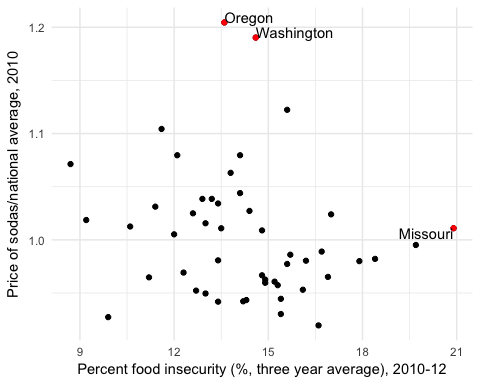
Next, clean the Food Prices and Taxes data frame (price\_df). The raw data looks like this:

## # A tibble: 3 x 11  
## FIPS State County MILK\_PRICE10 SODA\_PRICE10 MILK\_SODA\_PRICE10  
## <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 01001 AL Autauga 0.9703046 0.9722159 0.9232891  
## 2 01003 AL Baldwin 1.0176900 1.0013910 0.9401653  
## 3 01005 AL Barbour 1.1366710 0.9925836 1.0594000  
## # ... with 5 more variables: SODATAX\_STORES14 <dbl>,  
## # SODATAX\_VENDM14 <dbl>, CHIPSTAX\_STORES14 <dbl>,  
## # CHIPSTAX\_VENDM14 <dbl>, FOOD\_TAX14 <dbl>

Rename the SODA\_PRICE10 column, which gives the 2010 price of sodas divided by the national average by county. Rename the State column as well. Find the mean value of this statistic per state, and filter out states with missing soda price data. If you run the summary function on this data frame, it should look like this:

## state\_abb soda   
## Length:49 Min. :0.9195   
## Class :character 1st Qu.:0.9607   
## Mode :character Median :0.9890   
## Mean :1.0049   
## 3rd Qu.:1.0312   
## Max. :1.2045

Next, join together the food insecurity and soda price data frames by the state\_abb column, and plot the percent food insecurity versus the price of soda for each state. States with unusual values (Oregon, Washington, and Missouri) are highlighted. (*Hint*: Use these country names to to the subsetting necessary to create this highlighting. Join this subsetted data frame with the state\_names data frame to include full state names. To make sure the labels aren't directly over the points, use the vjust = "outward" and hjust = "inward" arguments.)



# Rates of obesity and diabetes

Next, clean the Health and Physical Activity data frame (health\_df).

## # A tibble: 3 x 14  
## FIPS State County PCT\_DIABETES\_ADULTS08 PCT\_DIABETES\_ADULTS13  
## <chr> <chr> <chr> <dbl> <dbl>  
## 1 01001 AL Autauga 11.4 13.0  
## 2 01003 AL Baldwin 9.8 10.4  
## 3 01005 AL Barbour 13.6 18.4  
## # ... with 9 more variables: PCT\_OBESE\_ADULTS08 <dbl>,  
## # PCT\_OBESE\_ADULTS13 <dbl>, PCT\_HSPA15 <dbl>, RECFAC09 <dbl>,  
## # RECFAC14 <dbl>, PCH\_RECFAC\_09\_14 <dbl>, RECFACPTH09 <dbl>,  
## # RECFACPTH14 <dbl>, PCH\_RECFACPTH\_09\_14 <dbl>

Rename the PCT\_DIABETES\_ADULTS13 column, which gives the adult diabetes rate in 2013 by county, and the PCT\_OBESE\_ADULTS13 column, which gives the adult obesity rate in 2013 by county. Rename the State and County column names as well. Select the two health outcome columns as well as state and county, then use the tidyr function gather to reformat the data frame to be more "tidy": there should be an outcome column with values for "Diabetes" and "Obesity", and a percent column with the value of each health outcome by county. The first few rows of the cleaned data frame should look like this:

## # A tibble: 3 x 4  
## state\_abb county outcome percent  
## <chr> <chr> <chr> <dbl>  
## 1 AL Autauga Diabetes 13.0  
## 2 AL Baldwin Diabetes 10.4  
## 3 AL Barbour Diabetes 18.4

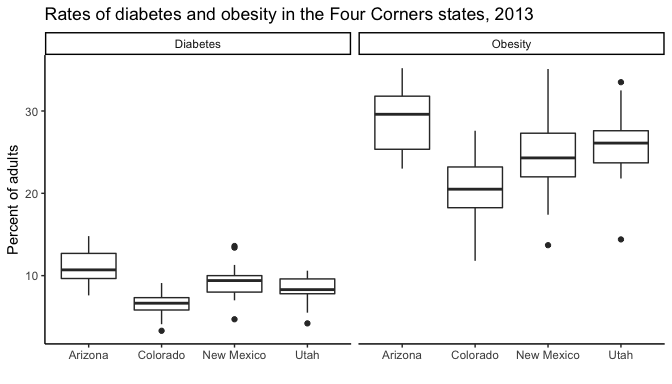
This table shows the average, minimum, and maximum values of obesity rates in 2013 for the five states with the highest mean obesity rate. ("*Hint*: Set na.rm = FALSE, or leave this as the default, in calculating all these statistics, so the final table will only include states with complete data across all counties.)

|  |  |  |  |
| --- | --- | --- | --- |
| State | Percent (mean) | Percent (minimum) | Percent (maximum) |
| Mississippi | 36.81 | 28.6 | 47.6 |
| Louisiana | 36.18 | 27.8 | 42.1 |
| Alabama | 36.13 | 27.4 | 46.3 |
| Arkansas | 35.90 | 30.1 | 45.5 |
| West Virginia | 35.44 | 28.8 | 42.1 |

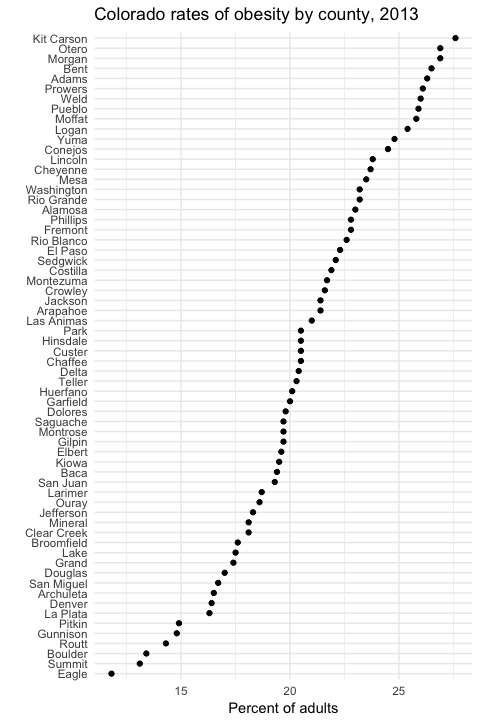
For our analysis of diabetes and obesity rates by county we'll focus on the Four Corners states: Arizona, Colorado, New Mexico, and Utah. Filter the cleaned Health data frame to include only these states, and then join this data frame with the state\_names data frame created above so that we have full names for each state. (Note: Depending on the order in which you join these data frames, your columns might be ordered differently than those shown here. If needed, you can use select to reorder your columns to match the order shown here.)

## # A tibble: 3 x 5  
## state\_abb county outcome percent state\_name  
## <chr> <chr> <chr> <dbl> <fctr>  
## 1 AZ Apache Diabetes 14.4 Arizona  
## 2 AZ Cochise Diabetes 10.3 Arizona  
## 3 AZ Coconino Diabetes 7.6 Arizona

This plot shows the distribution of health outcomes across counties in the Four Corners states in 2013, faceted to show differences in rates of diabetes and obesity.



The next figure shows the rates of obesity in each county in Colorado in 2013. Note that the points in this plot are arranged by the percent of obese adults in 2013. (*Hint*: Adjust the fig.height option in your code chunk so that the county labels are readable in the rendered figure.)



# Plotting function

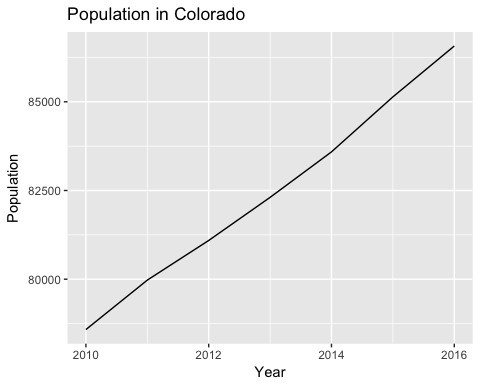
Finally, clean the "Supplemental Data - County" data frame (plot\_df), which gives population data by county from 2010 through 2016.

## # A tibble: 3 x 10  
## FIPS State County `2010 Census Population`  
## <chr> <chr> <chr> <chr>  
## 1 01001 Alabama Autauga 54,571  
## 2 01003 Alabama Baldwin 182,265  
## 3 01005 Alabama Barbour 27,457  
## # ... with 6 more variables: `Population Estimate, 2011` <chr>,  
## # `Population Estimate, 2012` <chr>, `Population Estimate, 2013` <chr>,  
## # `Population Estimate, 2014` <chr>, `Population Estimate, 2015` <chr>,  
## # `Population Estimate, 2016` <chr>

Rename columns at positions 4 through 10 to 2010, 2011, and so on through 2016, and rename the State column. Make this data frame more "tidy" by gathering to create year and population columns. Change the classes of year and population to integer class (use as.integer for this step). The first ten rows of the data frame should look like this:

## # A tibble: 10 x 3  
## state year population  
## <chr> <int> <dbl>  
## 1 Alabama 2010 71339.34  
## 2 Alabama 2011 71640.57  
## 3 Alabama 2012 71880.00  
## 4 Alabama 2013 72081.78  
## 5 Alabama 2014 72286.78  
## 6 Alabama 2015 72445.90  
## 7 Alabama 2016 72586.57  
## 8 Alaska 2010 24490.72  
## 9 Alaska 2011 24921.14  
## 10 Alaska 2012 25209.97

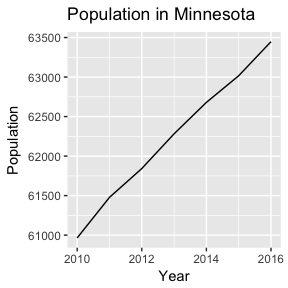
Finally, use this cleaned population data frame to write a function called plot\_pop that will plot population from 2010 through 2016 for a selected state. It has two arguments, state and df. If you run the function, it will subset data for a single state from the data frame specified by df and will plot year versus population for that state. For example, running plot\_pop(state = "Colorado", df = pop\_clean) would plot population over time in Colorado:



(*Hint*: Look closely at the plot and make sure your function creates a plot that looks the same, in terms of elements like axis and plot title. Also, you can assume that df, the data frame input to the function, always has a column named state with states and columns named population and year with the population for each year. You can also assume that the country name input for state is always a state that has data in the data frame, so you don't need to include any code for error checking.)

Here are a few more examples of running this function:

plot\_pop(state\_name = "Minnesota", df = pop\_clean)



plot\_pop(state\_name = "Illinois", df = pop\_clean)

