GEOG 215: Assignment #2

Continue Working with R and R Markdown

100 points

Introduction to Spatial Data Science

Paul Taillie

Updated: January 27, 2023

In this assignment, you will continue to work on your basic data management, processing, and displaying skills in R. You will be performing all your work within an R Markdown document for this assignment.

### Create a new R Markdown document

1. Open RStudio and make sure that your environment is empty, then choose File | New File | R Markdown. *Note: to clear your Environment, find the button that looks like a broom in the Environment tab!*

* In the popup window, enter “Assignment #2” for the title and for author, enter your name. Make sure that the type (on the left) is Document and the Default Output Format is HTML.
* Note that the new R Markdown document should have populated the “header” section to include the information you entered. For example, mine looks like this:

---  
title: "Assignment #2"  
author: "Paul Taillie"  
date: "1/27/2023"  
output: html\_document  
---

1. Task 1 is to change the header of your R Markdown document to modify the style of your knitted rmarkdown report. First, replace the auto-generated date with the inline R code below. This will automatically populated the date field with today’s date. Next, [go to this site and choose a “Pre-Packaged Theme”](https://www.datadreaming.org/post/r-markdown-theme-gallery/) and replace the word THEMEHOLDER in your header with the name of the theme you chose. [Go to this site and choose a highlight scheme](https://www.garrickadenbuie.com/blog/pandoc-syntax-highlighting-examples/) (and again replace the word HLHOLDER in your header with the highlighting style you chose).

* Also, it is important the “blank space” is correct in the header! To the left of html\_document is a single tab and to the left of theme and highlight is two tabs.

---  
title: "Assignment #2"  
author: "Paul Taillie"  
date: '`r format(Sys.Date(), "%B %d, %Y")`'  
output:  
 html\_document:  
 theme: THEMEHOLDER  
 highlight: HLHOLDER  
---

1. Note that a “new” R Markdown document in RStudio is populated with some example content to get you going. This can be useful if you forget how to do some basic things in Markdown. For now, you can just delete everything from ## R Markdown to ...generated the plot.

#### Did you know?

Your .Rmd file is saved every time you knit. However, it is also good practice to save occasionally if you’re spending a lot of time working between knits. You can do this by clicking the small disk button in the Editor panel.

1. Save your .RMD file using the following naming convention:

* **Lastname**\_GEOG215\_Assg**2**.Rmd
* For example, Dr. T’s file would be: **Taillie\_GEOG215\_Assg2.Rmd**

### Read in .csv file and summarize

For this assignment, we will be working with vaccination data from California for the years 2000 to 2015. Download the data from Canvas and save it in your GEOG215 project directory

The data you have downloaded is school-level data, meaning each row represents the values for a single, unique school. The columns of this table represent (for each school): an ID, name, geographic coordinates (in WGS84 coordinate system), school type (public or private), kindergarten enrollment (each year), and number of kindergartners with a Personal Belief Exemption from vaccination, PBE (each year).

1. Use read.csv() to read the data from where you saved it on your computer. Assign the name of the object as cavac (shorthand for “California vaccination”). This line of code should look like the command below (make sure that you change the location of the file to where you saved it in the download.file() command!). **Also, make sure to add a comment for this command!**

cavac <- read.csv("location/on/your/computer/filename.csv")

1. Under the code chunk you just created, insert a **third-level heading** to your R Markdown file named, “Data Summary” (without the quotes)
2. Copy and paste the text below into your **R Markdown** document below the heading you just created. Insert the correct function such that the values (not the code) are populated when you knit the document. If you’re struggling with this, see Step 21 in Assignment 1 or [this resource](https://r4ds.had.co.nz/r-markdown.html#inline-code).

The number of schools/rows in the table: `r `   
The number of columns in the table: `r `

### Calculate state-level values

Even when we have data separated into unique observations, it is often useful to examine the “study region” as a whole. In this case, we will aggregating the school-level data to the state-level and making some plots.

1. Under the code chunk you last created, insert a **third-level heading** to your R Markdown file named, “Aggregating data” (without the quotes)
2. First, we will aggregate (sum) each column (year) in cavac to get the “state” value, because all of the data set only contains observations in California. We could use the sum(cavac$column) function on each column, but it’s much easier to use the colsums(cavac) function once on the whole dataset. This is referred to as “vectorized programming” and much more efficient than applying a function several times (we’ll come back to vectorized programming later in the course). Add an R code chunk in your R Markdown document as follows:

```{r}  
## Sum the values in each column  
CA.sums <- colSums(cavac)  
```

1. Use the small green right arrow to run this code chunk. You should get the following error:

Error in colSums(cavac) : 'x' must be numeric

1. In the Environment window of RStudio (top right area), click the little arrow button next to cavac. Note that both the column labeled SCHOOL and TYPE are of type Factor (depending on your version of R, they might be type Character). Factors are categorical (nominal) data, thus we cannot perform mathematical functions on them (the same goes for data stored as type Character). In this case, we asked R to calculate the mean of a bunch of factor values and it (rightfully) did not like it because it’s impossible.
2. So now, we will perform the colSums function on only the columns that make sense (the ENR (enrollment) and PBE columns). First, we will print the column names to the screen and (in the next step) use table indexing to perform an action on a subset of an object. First, add the following command (and a comment!) above the colSums() command.

names(cavac)

1. Note the position of the first column that begins with ENR and the last one that begins with PBE. We want to sum these columns and all of those in between! Thus, we’ll use indexing as discussed in class to tell R which columns we want. Modify the colSums() command below replacing “X” and “Y” with the correct information. Basically, we are telling R that we **only** want to perform the function colSums on a portion of our data.frame. Click the little green right arrow to update the results.

```{r}  
CA.sums <- colSums(cavac[,X:Y])  
## Print column sums to screen  
CA.sums  
```

1. Uh oh… we still have a problem. In the Environment window of RStudio, notice there are some observations with NA values in the ENR and PBE columns, which stands for **No Data**. What this means is that the school/kindergarten was not open or did not report data that year! Unfortunately, NA values will affect some of our mathematical operations if not accounted for. In this case, the results of the colSums function contain only NA values.

#### Don’t forget to use the help files!

In the console, type ?colSums or find the Help tab in RStudio (mine is in the same pane as my file browser) and type colSums in the search box. These pages will provide you with information about R functions. In teh help file for colsums, you’ll notice there is an argument called “na.rm” where the default is set to FALSE.

1. To fix our issue with NA values, we will use a parameter/argument available in the colSums() function (as provided below) to tell the function how to handle NA values. In this case, we want to remove them from the summation for each column, so we add , na.rm = TRUE after the object in the colSums() function (see example below). Click the little green right arrow to update the results.

CA.sums <- colSums(cavac[,X:Y], na.rm = TRUE)

1. Now we have an object called CA.sums containing 32 values. The first 16 values are the statewide counts of children entering kindergarten in years 2000-15 and the final 16 values are the statewide counts of the number of children entering kindergarten with a PBE.
2. Now, we will create a new object containing the **Percent** of all kindergartners with a PBE for each year between 2000 and 2015. The command below uses R’s indexing ability. In this example, we can divide the two vectors by each other because the “years” line up for PBEs and Enrollment. The basic form of this is 100 \* PBEs / Enrollment. *Note: we multiply by 100 to convert a proportion to a percent value*. Insert the following commands in the prior code chunk (**along with comments!**). Click the little green right arrow to update the results.

CA.pbe.00.15 <- 100 \* CA.sums[17:32] / CA.sums[1:16]  
CA.pbe.00.15

1. Note that the output carried over the names from the prior object? This can get confusing, so it’s often worth it to rename observations and objects to reflect what they are. In this case, we use a command called paste(). There are a few lines of code provided that show you the basics of how paste() and paste0() work (**these should NOT be inserted in your R Markdown document, they are for reference only!**). Still confused? Take a look at the help file!

paste("first", "second", sep="-")  
paste("no", "separator", sep="")  
paste0("no", "separator")  
paste("vector paste ", 1:3, "!", sep="")  
paste0("different text ", 12:9, "?")

1. Insert the following commands in the prior code chunk (with comments). Click the little green right arrow to update the results.

paste0("PBER", 2000:2015)  
names(CA.pbe.00.15) <- paste0("PBER", 2000:2015)  
CA.pbe.00.15

1. We will now calculate the one-year change in the PBE percent for each year between 2001 and 2015. Note that we have 16 yearly values in the vector CA.pbe.00.15 (for years 2000 - 2015). We can only calculate 15 yearly change values because we do not have the value for 1999 (so we cannot calculate change for 2000). Again, we will use our ability to work with subsets of the data to accomplish this task by subtracting the prior year value for each year, e.g., the “change” value in 2015 will be the difference between the PBE percent in 2014 and 2015. Add the following code below to your prior code chunk (with comments!). Note that this code chunk has a few example calculations build to show you how this works.

CA.pbe.00.15[1] # year 2000  
CA.pbe.00.15[2] #year 2001  
CA.pbe.00.15[2] - CA.pbe.00.15[1] #year 2001 - year 2000  
CA.pbe.ch.01.15 <- CA.pbe.00.15[2:16] - CA.pbe.00.15[1:15]  
names(CA.pbe.ch.01.15) <- paste("PBERCH", 2001:2015, sep="")  
CA.pbe.ch.01.15

### Create plots

1. Under the code chunk you last created, insert a **third-level heading** to your R Markdown file named, “Plots” (<< without the quotes).
2. Under the Plots heading, create a new r code chunk. In this code chunk, create a basic plot of the geographic locations of all of the schools in California (use the cavac object for this). For these data, the column named **lon** contains the longitude (or X) coordinates and the column named **lat** contains the latitude (or Y) coordinates.
3. Now, you will create a line plot using the state-level data. I provided the basic plot() command to create a working plot, but you will need to provide the correct parameters. [Review “The basic plot command” section here (it’s very short and easy)](https://mathinsight.org/plotting_line_graphs_in_r) and make the following changes to the plot function in the code chunk! Alternatively, you may opt to use ggplot which we all know is better ;)

* Make the plot a **line** type
* Main title should be **PBEs in California, 2000-2015**
* X axis title should be **Year**
* Y axis title should be **PBE (%)**
* Line width should be **2**
* Line color should be **red**

plot(2000:2015, CA.pbe.00.15)

1. Spell check your .Rmd document (Edit | Check Spelling or the little button in RStudio with an “ABC” and check mark). Knit your document. **Review the knitted .html version of your assignment and look for issues/problems with formatting!** I suggest viewing the output in an actual browser rather than in RStudio’s browser (look for the Open in Browser button of the popup window with the knitted output).

### Deliverables

Upload your .Rmd and .html files to Canvas