

Practice Problems on Computing, Part 2

Write a script that solves the following problems. Save the script in the R subdirectory of your class folder. I recommend the format `mm-dd-title.R`, so maybe `09-14-computing-part2.R`.

1. Make sure you have a folder set up for this class as follows:

```
pols-209 (call this what you like, save it where you like)
|--data (important)
|--notes (optional)
|--R (important)
|--readings (optional)
```

You can call `class-folder` whatever you like (maybe `pols-209`), but I recommend keeping at least `data` and `R` folders named the same.

2. Download the data set `health.csv` from <http://www.carlislerainey.com/teaching/pols-209/data/health.csv> and place it in the `data` subdirectory of the class folder. (Just click the link and then File → Save As...) You can find out more about this data set at <http://www.carlislerainey.com/teaching/pols-209/data/health-codebook.html>. It is based on my recent [article](#) about the politics of the Medicaid expansion.
3. Point-and-click to set the working directory in RStudio to your class folder. (Hint: It's Session → Set Working Directory → Choose Directory...)
4. Use `read.csv()` to load the data. Remember that because you put the file in the `data` subdirectory and not in the main directory, you'll want to include that subdirectory as part of the path. That means the path will be `"data/health.csv"`, not `"health.csv"`. Store the data set as an object, and give it an informative name (maybe `health` or `health_data`?). In the questions that follow, I assume that you've assigned it to the object `health`.
5. Use `names(health)` to see the names of the variables in the data set.
6. Use `summary(health)` to see a numerical summary of each variable.
7. To get an even nicer summary, use the `glimpse()` function in the `dplyr` package. You'll need to install and load `dplyr`.
8. We haven't talked about how to do statistics yet—we're just now getting data loaded into R. But try `plot(health$percent_uninsured, health$percent_favorable_aca)`. What's going on in this plot?
9. Now install the package `ggplot2`.
10. Load the package `ggplot2`.
11. Now we're going to replicate the plot above with `ggplot2`, mostly to illustrate how packages can add functionality. Try `qplot(percent_uninsured, percent_favorable_aca, data = health)`. What is different than the plot above?
12. Use the `mean()` function to calculate the mean of the variable `percent_uninsured`.
13. Use indexing to find the fourth value of the variable `percent_uninsured`.
14. Turn the variable `percent_uninsured` into a proportion by dividing it by 100. Note that you can save this new vector as a variable in the data set by assigning it to `health$prop_uninsured` or simply save it to the environment by assigning it to `prop_uninsured`. You choose whichever you prefer.