Lab 6: Code peer review

Henry Bassett

September 16, 2021

1. Pair up!

Say Hello! If you know each other, catch up for a few minutes. If you don't know each other, introduce yourselves. What is your favorite season and why?

Use Campuswire to DM each other your `Lab03-wrangle-tidy.Rmd` file

> Note: *While GitHub makes it easy to share files, I want you to keep your repositories private/unshared with other students. So, please send each other an email with your `lab03_wrangle-tidy.Rmd` file attached.*

2. Review your partners lab with an eye on functionality, readability, and documentation.

Follow the coding assessment criteria for the calendar assignment to review your partner's code in terms of *functionality* (does it execute without error? is it reproducible on your machine? does it produce the expected output?), *readability* (is the code easy to read/follow? is the formatting consistent?), and *documentation* (are there any comments? are they useful/add value/help explain what particular code is doing?).

Provide comments in *this* `lab06` rmd file.

It is helpful if you refer to specific line numbers when referring to particular lines of code. It is beneficial to each of you to be critical in your review – better for your peer to point out errors, disorganization or confusions in your code now, then for the instructor to identify them in the calendar assignment!

Did your partner go about wrangling the data in the same way you had, or did they use a different "order of operations"? Did they use any functions you were unfamiliar with?

Remember to use appropriate formatting for inline code and code chunks.

3. Share your comments!

Knit this document with your comments, and respond to your partner's DM with this pdf. If you have extra time, discuss your respective strategies for tackling the wrangling with each other.

FEEDBACK

Functionality:

- *Line 169:* could be more concise by using `filter(!(Location %in% c("United States", "District of Columbia")))`
- *Line 242:* Consider creating a new dataframe, so you're not writing over the `healthcare` dataframe
- *Line 304:* this code tells us the amount of the spending for the highest state, but it doesn't give us information about what that state is. Prof. Bailey's answer in the solutions tells us what that state is.

Looks awesome! There are definitely some pieces of your wrangling code that are really good and I am going to steal :)

Readability:

- *Line 212:* consider saving the plot object in your environment (for example, `healthcare_plot <- ggplot(...)`)
- *Line 219:* add units to your axis labels (for example, "Employer Coverage (people)")
- *Line 278:* this is about your knitted PDF, but it's hard to read the numbers on the x-axis because they're all squished together; maybe move the legend to the bottom, or do something else to make the axis more readable
- *Line 305:* this is a really small thing, but make sure you have spaces on either side of the `==`
- *Line 307:* another small thing, but since the `max()` function is seperate from the wrangling you did in the previous line, there should probably be a line break here

Overall, the code is super easy to follow and read!

Documentation:

All the code is mostly documented in the questions!