Worksheet 10/14: Collaboration mini project I: shiny

MBio 691D

Lindsay Veazey

April 3, 2018

{shiny}

A few weeks ago, we practiced making interactive figures in R Studio using {leaflet} and {plotly}. Now we're building on that by using {shiny}, a nifty package that helps you display your interactive figures on webpages, R markdown documents (like this one), or even build dashboards (applications built with your particular data).

There are two parts to a shiny app: your code/UI (user interface) and the computer/server (that tells the web page how to read your code).

The best part about Shiny is that the developers set it up for folks who use R, but aren't software developers. Thanks to Evan for this great suggestion.

Important and cool Shiny feature: *reactivity*

Shiny employs *reactive programming*. This enables your outputs to *react* to changes in inputs. Basically, this means that when the value of variable x changes, then anything that relies on x gets instantaneously re-evaluated. This is very different from what we generally do in R, where variable assignments are static. Example:

```
x <- 5
y <- x + 1
x <- 10
```

In "regular" R, the value of y = 6. But in reactive programming, if x and y are reactive variables, then y = 11, because it would update with the updated value of x.

This powerful technique creates the responsiveness of Shiny apps. Only reactive objects variables behave this way; in Shiny, all inputs are automatically reactive. That's why you can always use input$x and know that whatever output you're creating will use the updated value of x.

This is a little weird because it's different, but here's a heads up for future reference.

+ GitHub

For this and the remaining classes, we'll go through various R tutorials in a collaborative way: I'll provide some editable R code in my 'potpourri' repo, and you'll share those edits

and additions via GitHub. This is the way things will go when you're collaborating on some code.

Half of you are "Part II", half of you are "Part III". If you are assigned to "Part II", navigate to https://github.com/lindsayveazey/potpourri (if you haven't already forked it from last week) or fetch origin in your GitHub Desktop app to update.

If you're assigned to "Part III", pick a partner and fork their repo to get a copy of "shiny.R".

Open "shiny.R" in R Studio; everyone will do Part I. Half of you will do Part II this week, the other half will do Part III.

Folks who do Part III will *pull request* your code edits with your partner at the end of class this week. Next week, you'll do the half you didn't do this week, with Part II folks using the helpful edits your classmates wrote into the code for you this week. We'll reverse the roles in a few weeks so everyone can pull request.

Got it?

PART I: Install {shiny} and being stylish

Refer to shiny.R to install the library and ensure it's working properly on your machine. You can try creating a template app two ways.

Make sure you save 'bclData.csv' to the same folder as your shiny template ('app.R').

You'll work with some helpfully simplified HTML coding (but R-ified). See this glossary for more explanation: https://shiny.rstudio.com/articles/tag-glossary.html or Shiny's developer site: https://shiny.rstudio.com/tutorial/written-tutorial/lesson2/.

And, a throwback to a prior worksheet- here's Google's R Style Guide: https://google.github.io/styleguide/Rguide.xml. As we go through the next sections of shiny.R, I'll intentionally make the code a little messy. Review the style guide and see if you can modify it to make it easier to read.

I'll throw in occasional hints in the inline comments.

PART II:

For this exercise, we're using 'bclData.csv' to construct our app. These data include information about the type, origin, flavor, name, alcohol content, and price of liquor products sold by stores in British Columbia.

Once you've completed Part II and your partner has completed Part III, commit and push your changes to your GitHub repo, then merge the changes from your partner's pull request.

PART III:

For this exercise, we'll loosely follow some of the many, many tutorials available on shiny.rstudio.com. We'll be shiny-izing the waiting time in between Old Faithful geyser eruptions.

If you're doing Part III this week, run through that portion of the code. Once you've made your edits, commit and push your changes to your GitHub repo, then pull request your Part II partner.

Once you've completed your pull request, go to http://www.shinyapps.io/ to sign up for an account. Follow the directions and publish your geyser app.

Review:

- You worked through a Shiny tutorial and built your first app.
- You pull requested or merged changes with a partner.
- You cleaned up some code based on Google's R Style Guide.

Here's a handy cheatsheet: https://shiny.rstudio.com/images/shiny-cheatsheet.pdf There's some unfamiliar stuff there, so it may be only somewhat helpful at this point, but there it is for future use.