Reading Set 3b: Shiny Tutorial Video

Ziji Zhou

Due by 10pm ET on Monday

Reading Set Information

This week we will replace reading with watching a tutorial video!

Getting practice with the code while watching the video prior to class will allow us to jump into things more quickly in class and dive deeper into topics. As you actively watch the tutorial, you will work through the Reading Set to help you engage with the new concepts and skills.

These should be completed on your own without help from your peers. While most of our work in this class will be collaborative, it is important each individual completes the active readings. The problems should be straightforward based on the textbook readings, but if you have any questions, feel free to ask me!

GitHub Workflow

1. Before editing this file, verify you are working on the copy saved in *your* repo for the course (check the filepath and the project name in the top right corner).
2. Before editing this file, make an initial commit of the file to your repo to add your copy of the problem set.
3. Change your name at the top of the file and get started!
4. You should *save, knit, and commit* the .Rmd file each time you've finished a question, if not more often.
5. You should occasionally *push* the updated version of the .Rmd file back onto GitHub. When you are ready to push, you can click on the Git pane and then click **Push**. You can also do this after each commit in RStudio by clicking **Push** in the top right of the *Commit* pop-up window.
6. When you think you are done with the assignment, save the pdf as "*Name_thisfilename_date*.pdf" (it's okay to leave out the date if you don't need it) before committing and pushing (this is generally good practice but also helps me in those times where I need to download all student homework files).

Gradescope Upload

For each question (e.g., 3.1), allocate all pages associated with the specific question. If your work for a question runs onto a page that you did not select, you may not get credit for the work. If you do not allocate *any* pages when you upload your pdf, you may get a zero for the assignment.

You can resubmit your work as many times as you want before the deadline, so you should not wait until the last minute to submit some version of your work. Unexpected delays/crises that occur on the day the assignment is due do not warrant extensions (please submit whatever you have done to receive partial credit).

Additional Instructions

The Shiny Tutorial video is a recording from an older webinar series, so the slides and code are not at the bitly link provided and you can ignore references to that, and obviously the dates of the webinar are meaningless.

I have included all the code and slides from the tutorial in the folder for Reading Set 3 (make sure you have copied over the entire folder with all its contents!). I obtained the files from the corresponding Shiny tutorial repo.

Use the **Files** pane in RStudio to navigate to the individual R scripts in tandem with the video tutorial to run the code along with the speaker.

There are 3 parts to the Shiny tutorial, and all 3 parts are in the same 2.5-hour video. You can power through all three in one sitting, or I've embedded time-stamped links in this document that will allow you to jump to Part 2 and Part 3 when you're ready, if you want to break up the tutorial a bit.

Keep track of your questions, big and small, as they come up while you are coding along with the video and taking notes. It's possible a question you have early on will be answered as you continue watching the video. Any remaining questions you can do some answer-finding after the video or post to Campuswire if you still can't figure something (or some things) out.

Finally, you should *NOT* publish your app with shinyapps.io for this Reading Set. We will use class time this week to publish our first app.

Part 1 **How to build a Shiny app (25 – 30 minutes)** Watch Part 1 of the tutorial. After the demonstration of the second shiny app script (`02-hist-app.R`), skip over the "Share your app" sections and jump to the 40-minute mark) to watch the brief recap of Part 1. Feel free to modify either the R script files themselves or to copy and paste the code into this document to get practice.

1.1 What are the 3 main components needed to build a shiny app, and what is the purpose of each?

The three components are the UI, the server, and the R script. A shiny app requires a UI where the user interacts with the inputs which then is sent to a server running the R script. The server will run the R script in response to the changed inputs and generate a new output to send to the UI again and display the output.

Part 2 **How to customize reactions (50 – 60 minutes)** Continue on to Part 2 of the tutorial (starting at 43:32).

2.1 What is the general idea behind *reactivity* for a Shiny app?

Reactivity is the the interaction between the input of the user and the output coming from the R server. Integrating reactivity into Shiny apps allow for more layers of data presented and a level of customization in output for the user.

2.2 Give an example of how you might use one (or more) reactive value(s) in conjunction with a reactive function.

The `renderPlot()` function can allow me to assign reactive values inside of it in order to change elements of the plot within the render function. For example, in the UI section of the Shiny app I can include a slider getting a value between 1-100. That number would be stored within as `input$num`. Then, within the output the `renderPlot()` function will include a `ggplot` that includes an element of `scale_x_continuous(n.breaks = input$num)`, allowing the user to input the number of ticks on the x axis to customize their graph. Considering the change in ggplot labeling won't require a change in the actual datapoints in the graph, the change will exist wihtin an `isolate()` function.

Part 3 **How to customize appearance (40 – 50 minutes)** Continue on to Part 3 of the tutorial (starting at 1:33:41). Skip over the *Style with CSS" section, and jump to the 2:24:35 mark to wrap up the webinar!

3.1 What is one question that still remains after completing the webinar (and after trying to answer yourself with your answer-finding skills)?

I think it is less of a specific answer but rather the intricacies of formatting the web page dynamically vs. a fixed page. With the variety of screen sizes and ratios the criteria to build a functional page for all platforms (or even simply a few of the most popular ones) seem difficult with the unique intricacies each platform presents.

Part 4 **On your own** Follow the steps below to create a new shiny app.

4.1 Create a new folder in this directory called "rs3_shiny" (i.e., the filepath will be `YOUR-REPO/problem-sets/rs3/`

4.2 Open a new *R Script* file (*not* R Markdown) by going to **File** → **New File** → **R Script**. R Script files only take R code and comments (anything you would put in a code chunk).

4.3 Save the file as `app.R` within the "rs3_shiny" folder.

4.4 Copy the code from the `part-2-code/02-two-outputs.R` file and paste it into your new `app.R` script.

4.5 Add a `textInput()` widget that allows the user to change the title of the histogram (similar to the code in `part-2-code/01-two-inputs.R`). Update the code in the `server()` function appropriately. Run the app to make sure it works as you expect.

4.6 Update the layout of the app to use a `navlistPanel()` structure (similar to the code in `part-3-code/06-navlist.R`). *Hint*: put `navlistPanel()` around the output objects only.

4.7 Make sure the app runs successfully, then save your changes in the `app.R` file, and push the contents (the folder and its contents) to your GitHub repo.