Prep4S24

BilalTariq

2024-02-22

Reminder: Prep assignments are to be completed individually. Upload a final copy of the .Rmd and renamed .pdf to your private repo, and submit the renamed pdf to Gradescope before the deadline (Sunday night, 2/25/24, by midnight).

Reading

The associated reading for the week is Chapter 14. The prep involves looking through one tutorial and working through another - estimated time at least 20 minutes each.

Note: There is no practice set this week, so that you can focus on completing the calendar query individual project due Wednesday by midnight and start on the Shiny group project, where an initial proposal is due Thursday by midnight.

1 - Exploring Shiny

A Shiny app is an interactive visualization. These apps can be written in R using the *shiny* R package. We want to explore Shiny and its many uses with this prep.

To get a sense of what's out there with Shiny, let's look at some example tutorials.

Stir your browsers to this tutorial: http://zevross.com/blog/2016/04/19/r-powered-web-applications-with-shiny-a-tutorial-and-cheat-sheet-with-40-example-apps/

This is a tutorial on (fairly simple) Shiny apps which are each created from a SINGLE file. For larger apps, you would want to create separate server.R and UI.R files, the first one governs the R computations and the second one governs the UI. There are many examples of these files online.

When tackling learning Shiny, don't start from scratch! Find an example that does some of what you want and take its code and modify it.

Let's look through some of the tutorial now. For each example app, there is a code file showing the R code and an App tab that shows what the resulting app looks like. Note that these do not compile like .Qmd/.Rmd files. You RUN the apps instead of knitting. Alternatively, you'd make an R script file instead, and RUN within that.

Take a look at a few of the first apps and their code. Most of these are skeletons to demo the structure of Shiny.

The apps get more complicated as the tutorial proceeds. Try to look through the commands and see how the apps can do different things. Then, answer the following questions about some of the examples. Note: even though this directs you to only some of the examples, you will really benefit from looking through them all.

(If you want details about the data set, remember you can look up its help file.)

> part a - Take a look at Code 30/App 30. What does this App let you do with the mtcars data set?

Solution: This app displays a data set that is interactive. Firstly, using the slider we are able to adjust the number of rows that we want in the data set. Secondly, we are also able to search for which car we want and also for its mpg, disp and cyl. We can also sort by ascending or descending order according to mpg, disp or cyl.

> part b - What about Code 31/App 31?

Solution: Using PlotLy, we can see an interactive plot with vast functionality. There's a slider to alter the scale of the plot, and there is a toolbar on the top right corner of the plot that allows us to zoom, pan, download as a png and select alongside various other functions that would not typically be available on a standard ggplot.

part c - What about Code 33/App 33?

Solution: We have a slider that changes the number of points on the map and we can interact with the map by scrolling and grabbing with our mouse to pan. (There are also +/- buttons that can be used to zoom).

part d - What about Code 37/App 37?

Solution: We can interact with the plot by clicking on one of the points and it shows us that specific point in a table format at the bottom. We can also box select a bunch of points whose data is displayed at the bottom. There is an option to select how many rows we want displayed as well.

part e - One last one! What does Code 40/App 40 let you do with the mtcars dataset?

Solution: There is an interactive UI with an option to switch between a dashboard view and a widgets view. On the dashboard view there are two plots; one of which shows count v.s mpg and the other shows a linear regression of wt v.s mpg. Both of these can be altered by altering the slider that is at the bottom of the screen which changes the scale.

The blog entry is just one of many tutorials about Shiny out there.

2 - One more tutorial

The main home page for Shiny is https://shiny.posit.co/

It has a tutorial page for R that sends you to Lesson 1 (Shiny Basics - Part 1 - Welcome to Shiny at https://shiny.posit.co/r/getstarted/shiny-basics/lesson1/index.html

Working through a good bit of this tutorial will be part of the upcoming Shiny project. For now, we'll just look at the first part - Welcome to Shiny - to get an example app running. (As soon as the second project assignment is ready to post, I'll post it and you could start on the individual part working through the rest of the tutorial, etc.)

The aim here in "Welcome to Shiny" is for you to modify an existing Shiny app (provided) and run it! You may need to install the Shiny package if you are working on your own computer, if you do not already have it, before getting started.

You can *run* a shiny app in RStudio on your computer; you do not need to work on the server to run the apps. However, eventually, we will need to publish our apps to allow others to view them. You can do this either through the R server or through external sites such as shinyapps.io. We will look at this in lab. For the prep, you do not need to publish the app.

Read through the page "Welcome to Shiny". When the tutorial says its "Your Turn" and requests you make some changes to the app.R file, do the following:

1. Be sure you have the app.R file that is posted for the Prep. Copy it into your repo if you didn't already. You don't have to put it in a sub-folder called App-1 like they say, but you can if you want. The idea is just to be sure you have the app file ready to go.
2. Run the app. It should show the same starting point as the tutorial app.
3. Make the requested changes/updates to the app in the tutorial.
4. Run the app with the changes and be sure it matches the tutorial.

Then, complete the questions below, working through the rest of "Welcome to Shiny".

> part a - Were you able to implement the requested changes and run the updated app?

Solution: Yes I was able to implement all the changes and run the updated app.

When you get to "Go Further", please look at at least 5 of the other pre-loaded examples in the Shiny package.

> part b - What pre-loaded example did you find the most interesting?

Solution: I found the Movie explorer highly interesting as it was very interactive and crystallised all of the movies.

part c - Are there any applications you can think of where you might like to investigate the use of a Shiny app?

Solution: A shiny app would be very suitable for interacting with a dataset of the most common Majors at Amherst and their outcomes.

(Note, the next project will require you to work with an assigned group and make a Shiny app, so this is preliminary brainstorming about your individual interests.)

Finally, if you are interested in more information about Dashboards, or more examples of Shiny apps, here are some reference pages hosted by RStudio.

https://rstudio.github.io/shinydashboard/

https://shiny.posit.co/r/gallery/

https://shiny.posit.co/r/gallery/#user-showcase