Interactive Dashboards with R and Shiny

Steven Nydick Fiona Lodge Yu-Ann Wang

April 29, 2022

Setup

1. Requires
   - https://www.r-project.org/
2. Suggests
   - https://www.rstudio.com/products/rstudio/download/
3. Materials
   - Git: https://github.com/swnydick/siop-2022-interactive-shiny
   - Folder: https://github.com/swnydick/siop-2022-interactive-shiny/archive/refs/heads/master.zip

Package Installation

This demonstration goes over functionality of the following packages:

- htmltools
- shiny
- ggplot2
- thematic
- bslib
- shinyWidgets

Package Installation

This folder uses the `renv` package. To set things up, simply connect to the internet, open the project in RStudio, and then run the following line of code:

```
renv::restore(prompt = FALSE)
```

If you do not want to use RStudio, you need to make sure you are in the correct directory and then run the `renv::restore(prompt = FALSE)` line of code.

If you do not want to use `renv`, you will need to install everything manually (for example):

```
install.packages("shiny")
install.packages("shinyjs")
```

And then load them with the following line of code (for example):

```
library(shiny)
library(shinyjs)
```

What we want from you:

1. This session assumes little familiarity with `R`. If you have questions, please ask or email the presenters. We have used `R` for a very long time and sometimes forget what we had to learn.
2. Try to run all of the code in `RStudio`. The setup of the demonstrations naturally works in `RStudio`. If you do not have `RStudio`, you can certainly run all of the code in `R` or a different IDE.
3. Have fun!

Reactivity

Shiny uses `R` and Javascript tricks to create reactivity:

- Clicking on a UI element triggers other elements to evalulate.
- Updating a plot or a table can create feedback for other input/display objects.
- Code no longer appears to run in a standard linear order. Things later in the code trigger things earlier in the code.

These tricks make it easy to create complex, custom, and very flexible applications, but they require thinking in a way that is different from typical R programming...

Structure of Tutorial

The following is the general structure of the tutorial. Each of the parts has additional descriptions as to new functions and logic.

1. Using shiny to display HTML (static UI)
2. Adding input controls (dynamic UI/text generation)
3. Rendering dynamic plots based on input
4. Creating and using reactive functions/values
5. Linking reactivity across many tables/plots in an app
6. External packages to enhance functionality
7. Easily adding custom themes without CSS
8. Adding custom javascript elements/buttons
9. Encapsulating shiny logic within modules
10. Keeping in mind pitfalls of reactivity

Exercises