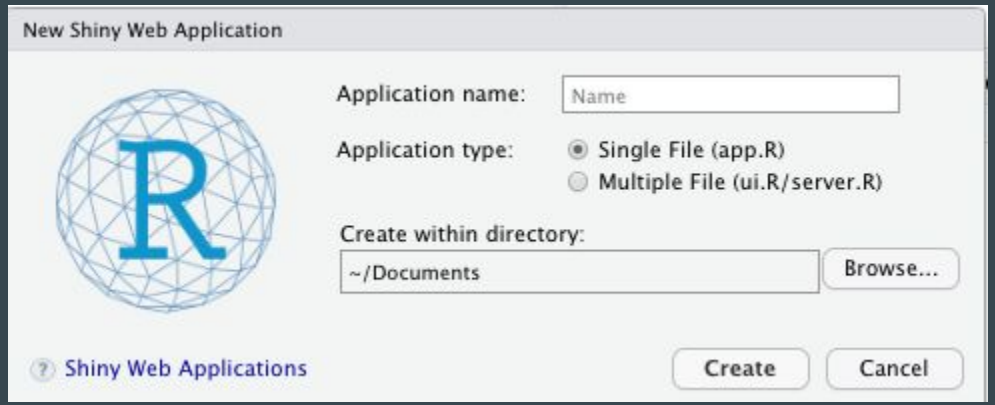
R Shiny Intro

Brittney Hernandez

Getting R Shiny on your computer

- Need RStudio
- I'm running R version 4.0.2 and R Studio version 1.3.1056
- Install and call the shiny package to the library
  - install.packages("shiny")
  - library(shiny)
- File > New File > Shiny Web App...



Shiny Intro

- Three main components:
  - ui { }
  - server { }
  - shinyApp(ui = ui, server = server)
- In the past these were saved separate script files (ui.R and server.R), but can now be defined in a single script file called app.R and shinyApp( ) will bind the ui { } and server { } elements
- Separate ui.R and server.R scripts can be useful when the app is complex and contains a lot of code

Shiny Intro

- An optional component is css { } or css.R which can be used to design and format the app
  - There are Shiny themes which can be used in place of the css component
    - Install.packages("shinythemes")
    - library(shinythemes)
  - If not using a theme you can use css { } in the app.R file
  - Or css.R which needs to be located in a www/ folder in your working directory
- The www subfolder contains any optional elements e.g., the CSS, images, .js, etc.)

app.R

- ui { } defines the user interface.
  - In the user interface you will specify “placeholders” for any content that you want the user to see or engage with
  - For example, in the ui, I will specify what the page header looks like and the layout of the app. If I want users to navigate to a new page by clicking a button, I will create a placeholder for the button and design it’s appearance in the ui
- server { } gives instructions for building the ui
  - The server is where any actions, reactions, displays will be specified in the server. For the button I created in the ui, the server is where I will then specify what happens when the button is clicked.

Terminology

- Page: a page or tab in an application
- Panels, rows and columns: determine the layout of a page
- Module: a piece of a shiny app code that can represent input, output or both
- Widget: web element that users can interact with

Basic widgets

Buttons

Action

Submit

Single checkbox

- [x] Choice A

Checkbox group

- [x] Choice 1
- [ ] Choice 2
- [ ] Choice 3

Date input

2014-01-01

Date range

2017-06-21 to 2017-06-21

File input

Browse... No file selected

Help text

Note: help text isn't a true widget, but it provides an easy way to add text to accompany other widgets.

Numeric input

1

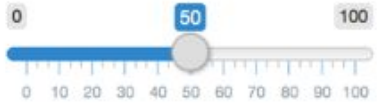
Radio buttons

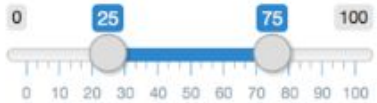
- (•) Choice 1
- ( ) Choice 2
- ( ) Choice 3

Select box

Choice 1

Sliders





Text input

Enter text...

Things I wish I knew when I first started...

- Each module is treated as its own local environment. So, objects defined by = or <- will not carry across modules
- To call an object from a different module, use <<- which creates a non-local assignment (almost global)
  - See the following link for more info about scoping in R
- There is a lot of nesting that happens within the respective ui and server elements so parentheticals can quickly become overwhelming
  - I recommend finding a system to help you signal the open and close of parentheticals
  - I like to use comments to define closing parentheses, brackets, etc.

Onto an example...

- The final code for the example can be found in the Shiny Presentation Dropbox folder but we're going to work up from the default code
  - First open the project: Shiny Presentation > Shiny Presentation.Rproj
  - Then the app itself: Shiny Presentation > example > app.R
- We'll work through replicating a plot from fivethirtyeight.com of Joe Biden's approval ratings
- When opening a new Shiny App, the app.R file by default includes app code for interactively choosing bins for a histogram plot. There's more info about this in Shiny's instructional articles
  - You can also use the function reference to see the specific elements needed for different Shiny functions

Designing the layout of the ui

- There are multiple ways to set up the ui layout
- The two we'll look at are the sidebarLayout and fluidPage
  - sidebarLayout sets up a sidebar on the side of a main panel
  - fluidPage uses a grid layout and you can determine where an element goes by using column (this is my favorite option)
- As mentioned there are themes we can be used design the look of the app
- Anywhere there is text, we can use html code to format - otherwise it will format according to the theme defaults

Adding widgets

- Shiny has some basic control widgets but there are also packages you can install to include more complex widgets (e.g., sortable, a widget to drag and drop inputs)
- The example adds a button and a select box

Adding images

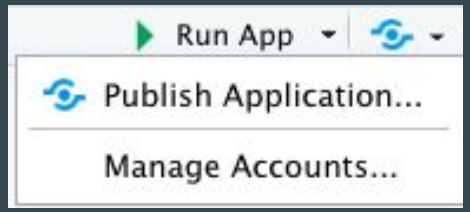
- Including images: Photo needs to be in a folder called www in the working directory
  - Use getwd() to see your working directory
- The code I use can only work with .png files and not .jpegs
  - I'm not sure exactly why but haven't looked into it bc I've found it easy to convert a .jpeg to .png

Running and viewing your app

- The dropdown next to Run App lets you run the app in different ways:
  - Run in Window
  - Run in Viewer Pane
  - Run in External - will open a web browser
- Can test results results by printing to the console
  - print(input$sample1)

Deploying an app

- The easiest way to deploy an app is through shinyapps.io but there are other options
  - See this link for other options
- Create an account with shinyapps.io
- After the account is created, go to Account > Tokens > Show
- This will give you a line of code to add to your app, authorizing your shinyapps.io account
- Publish the app using Publish Application



Using Shiny apps as a survey

- You can collect data from Shiny and use the app as a survey by connecting it to Dropbox or a local drive and saving the data as an .xlsx file.
  - Uses the package rdrop2
  - Create Dropbox folder to store data
  - Import a file into R to store data > Bind new case > Resave file
    - For the first case we need to create a new file. Create a condition looking for a file, and if it doesn't exist create it
    - To import an excel file into R, it needs to contain a minimum of 5 cases so when we create the dataset it will contain 5 rows of NAs which will be replaced as we collect data