Shiny II

INFO 201

Today’s Objectives

Review Shiny structure

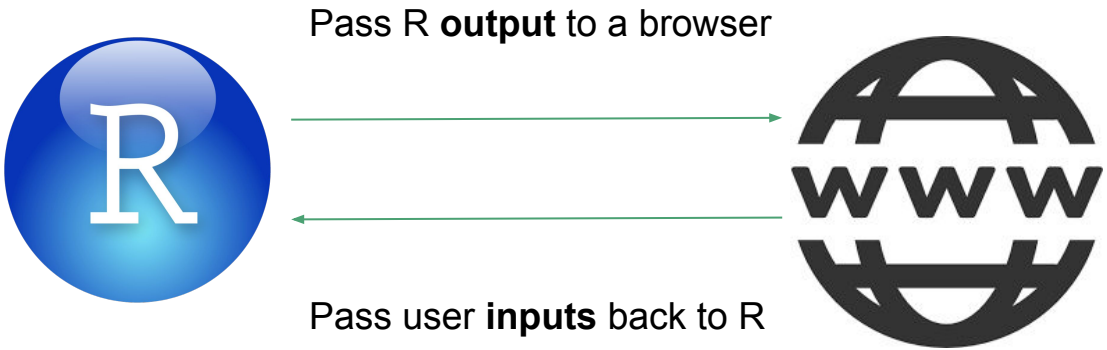
Practice using different widgets

Use Plotly with Shiny

Create UIs with multiple tabs

Review


Shiny
by RStudio
A web application framework for R


Pass R **output** to a browser
R
WWW
Pass user **inputs** back to R

Widgets

Widgets

Set of elements that a user interacts with

Send information back to server.r as input$VALUE

Facilitates interactivity

```r
# ui.R file ------------------------------------------------------
shinyUI(fluidPage(

  # Create a text input element
  textInput("text", label = h3("Text input"), value = "Enter ..."),

  # Show output$userText from server.R
  textOutput('userText')

))

# server.R file ------------------------------------------------------
### Use input to create a string
shinyServer(function(input, output) {

  # Render some text, using the value from the ui
  output$userText <- renderText({
      return(paste0('The user typed: ', input$text))
  })
```

Simple UI Widget

```
# Add a remote called "upstream"
git remote add upstream https://github.com/INFO-201/m14-shiny.git

# Pull in master branch changes from upstream remote
git pull --rebase upstream master

# Pull in complete branch changes from upstream remote
git checkout complete
git pull --rebase upstream complete
```

Rebasing changes from upstream remote (if you already forked/cloned module-14)

module 14 demo-2

module 14 demo-3

Plotly + Shiny

Plotly + Shiny

Play pretty nicely together

Enables better (interactive) graphics

Leverage Plotly-specific functions for rendering plots

```r
# ui.R file ----------------------------------------------------
shinyUI(fluidPage(
  mainPanel(
    plotlyOutput('scatter'),
    plotlyOutput('map')
  )
))

# server.R file ------------------------------------------------
shinyServer(function(input, output) {

  # Render a plotly object
  output$map <- renderPlotly({
    BuildMap(data, input$export)
  })
})
```

Using Strings as Variables Names

Why would we want to do this?

Interactively choose variables to use in analysis/charting

Doable (but a bit unpleasant):

```
life.expectancy <- c(75, 73, 71)
var.to.graph <- 'life.expectancy'
# Parse and evaluate var.to.graph
eval(parse(text = var.to.graph))
[1] 75, 73, 71
```

module 14 exercise-3

Layouts and Tabs

Layouts and Tabs

Shiny supports multiple interface **layouts** (i.e., sidebarLayout)

You can arrange your content into different **panels** (i.e., side, main)

Create multiple pages with **tabs**

Navigate across tabs in a navigation **page** (i.e., navbarPage)

navbarPage

tabPanel

Electoral College | Map | Scatter

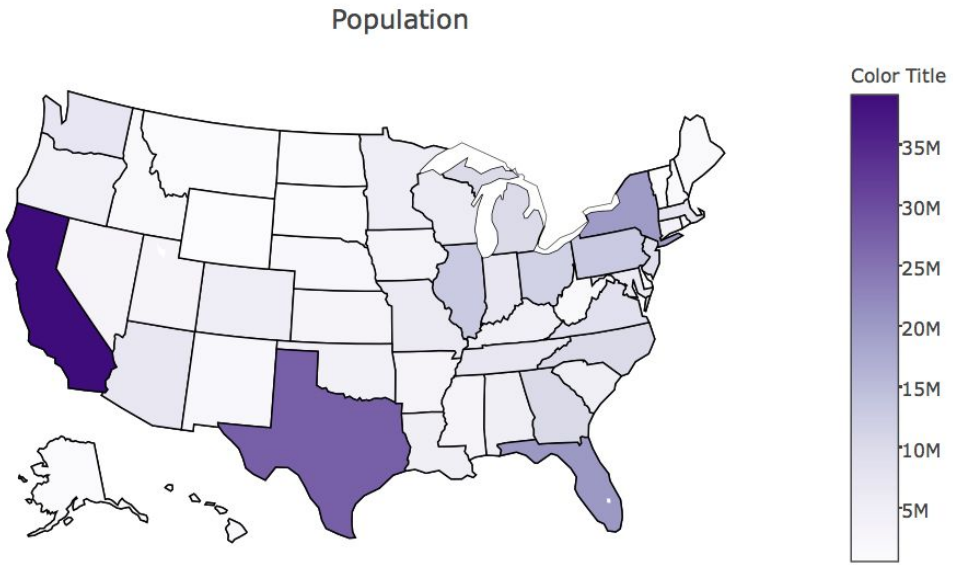
sidebarPanel

Electoral College Votes

Variable to Map

Population

mainPanel



Exercise 4

```
shinyUI(navbarPage('Electoral College',
                   # Create a tab panel for you map
                   tabPanel('Map',

                            # Create sidebar layout
                            sidebarLayout(

                               # Side panel for controls
                               sidebarPanel(

                                 # Input to select variable to map
                                 selectInput('mapvar', label = 'Variable to Map', choices =....)
                               ),

                               # Main panel: display plotly map
                               mainPanel(
                                 plotlyOutput('map')
                               )
                            )
                   ),
                   ......
```

module 14 exercise-4

Upcoming...

By Thursday: Be confident with **module 14**

Due Tuesday, 11/22 (***before class***): a8-building-apps

No lab this week

No office hours Thursday

Awesome guest lecture Thursday