Making the Chicago Cubs *Shiny*

Gage Sonntag

MMA '18 Queens University

Agenda

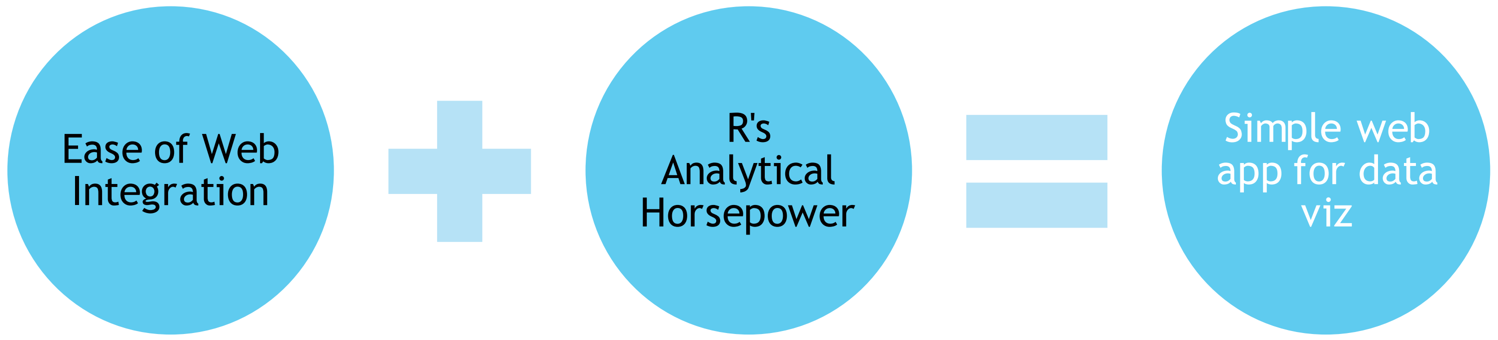
- Today's Goals
- What is Shiny?
- Some Shiny Examples
- Designing Shiny
- The Cubs Dataset
- Building our UI
- Building our Server

Today's Goals

- Understand what Shiny is, why it is a useful tool for your toolbox
- The structure of a shiny app
- Web Scraping
- Build a simple shiny app to look at data on the Chicago Cubs

What is Shiny?

- A web application framework for visualization



- Well Supported community
  - https://shiny.rstudio.com/

Some Shiny Examples

- A great gallery of examples, including code!
  - https://shiny.rstudio.com/gallery/
- Can do simple exploratory analysis
  - https://shiny.rstudio.com/gallery/movie-explorer.html
- Can be used for automated reporting
  - http://shiny.datacamp.com/rmarkdown-apps/rmarkdown_1.Rmd
- Or utilize some interesting features of ggplot
  - https://gallery.shinyapps.io/college_explorer-master/

Git some data to visualize

- Lets scrape data to visualize from Baseball Reference
  - https://www.baseball-reference.com/teams/CHC/2016.shtml
- Fork my repository that has already done the hard work
  - https://github.com/16gs24/MMA-2018-Shiny-Workshop

Designing Shiny

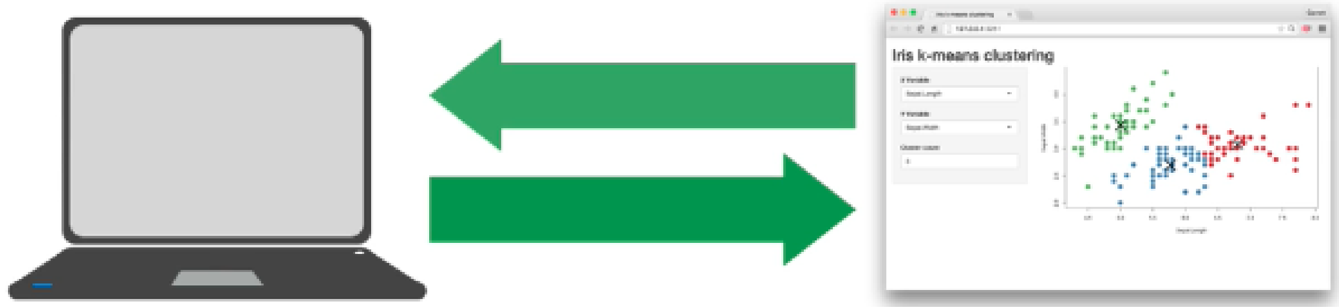
Server

- Provides code to convert our inputs to outputs
- Filters and transforms data into summary statistics a plot

UI

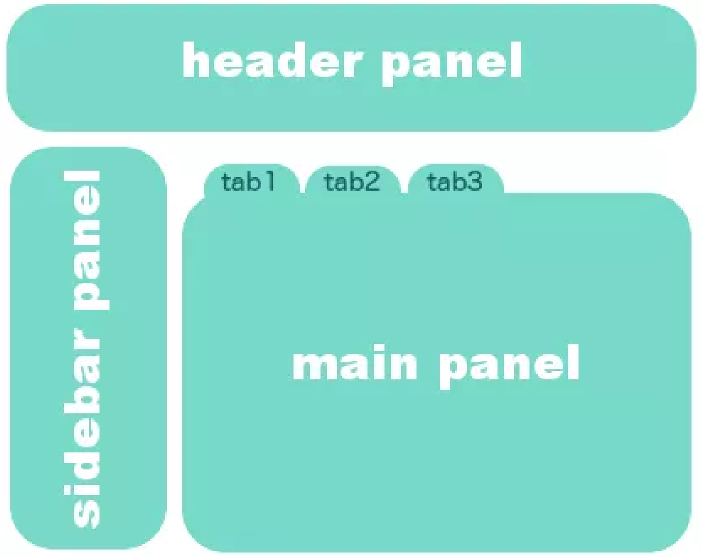
- "Control Widgets" are what the user interacts with
- The order of things on the screen

Every Shiny app is maintained by a computer running R



UI

- The framework for where objects appear
- Consider this HOW we select our inputs and WHERE our outputs get rendered
- One potential direction:

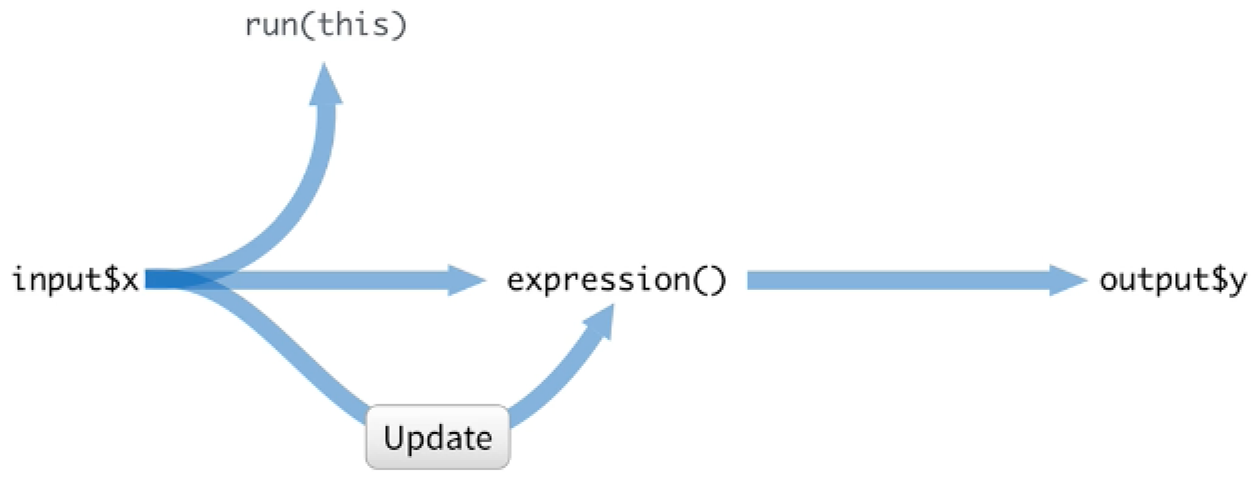


Server

- The control link between our inputs and outputs
- Three Rules:
  - Save outputs as output$
  - Render the outputs with render*()
  - access inputs with input$

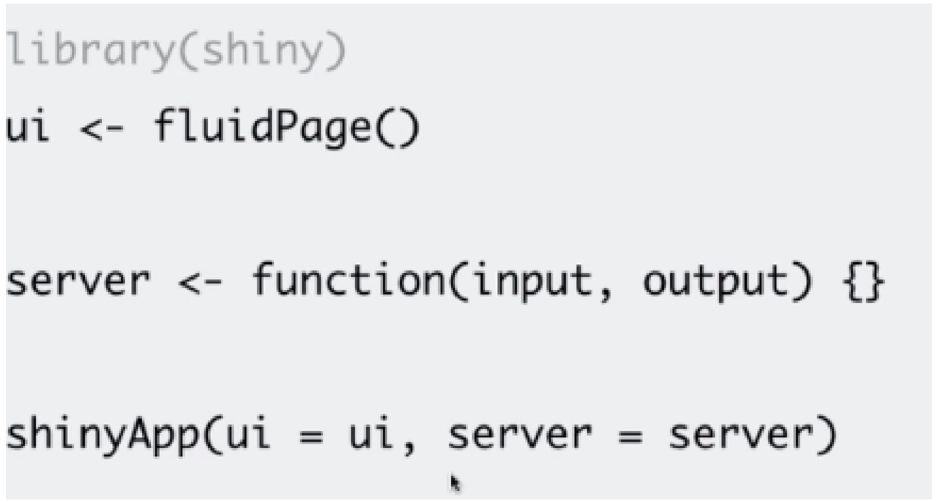
Sidenote: Terminology

- Reactivity: Shiny can dynamically change our output as you adjust inputs, or wait until you select a button. Good for drop downs, bad for text boxes.
- FluidPage: Makes our layout reactive to the dimensions of the web page



Putting It All Together

- Begin with a template:



```
library(shiny)
ui <- fluidPage()

server <- function(input, output) {}

shinyApp(ui = ui, server = server)
```

- Build out with consideration of INPUT & OUTPUT

Summary

- In a few short hours we:
  - Pulled data from the web
  - Filtered, & feature engineered it.
  - Built some visualizations to explore why people come to baseball games