Intro to Shiny (outline)

Martin Frigaard

2022-01-09

# Overview

Shiny gives us the ability to convert R code into an interactive application, without having to learn HTML, CSS or JavaScript. Shiny applications are incredibly helpful for sharing analysis results with audiences who might not be able to read R. This tutorial will cover an introduction to shiny architecture, common layout options, and application deployment.

We will also cover flexdashboard, a package for creating dashboards built with rmarkdown. We’ll close by demonstrating how to convert a flexdashboard to a shiny application.

## Shiny app anatomy

* ui
* server
* app.R

## User Interface (UI)

* layouts
  + [fluidPage()](https://shiny.rstudio.com/reference/shiny/1.4.0/fluidPage.html)
  + [shinydashboard](https://rstudio.github.io/shinydashboard/)
* input options
  + [sliderInput()](https://shiny.rstudio.com/reference/shiny/1.6.0/sliderInput.html)
  + [selectInput()](https://shiny.rstudio.com/reference/shiny/1.6.0/selectInput.html)
  + [checkboxInput()](https://shiny.rstudio.com/reference/shiny/1.6.0/checkboxInput.html)
  + [textInput()](https://shiny.rstudio.com/reference/shiny/1.6.0/textInput.html)
  + [textAreaInput()](https://shiny.rstudio.com/reference/shiny/1.6.0/textAreaInput.html)

## Server

* building reactive()s
* linking inputs and outputs
  + render\_\*() functions

## flexdashboard

* Recap rmarkdown
* What belongs in a dashboard?

## flexdashboard Layouts

* Sidebars
* Columns
* Rows
* Pages
* Tabs

## shiny + flexdashboard

* Converting from flexdashboard to shiny app
* Combining shiny reactivity with flexdashboard

## Packages

* inspectdf package
  + graphs, syntax
* reactable package
  + table displays

## Themes

* Bootstrap themes