R SHINY

Building Interactive Web Applications with R

JEREMY BUHLER, DATA LIBRARIAN
ALBINA GIBADULLINA, GRADUATE ACADEMIC ASSISTANT

UBC Library Research Commons

WORKSHOP PREREQUISITES

Software requirements:

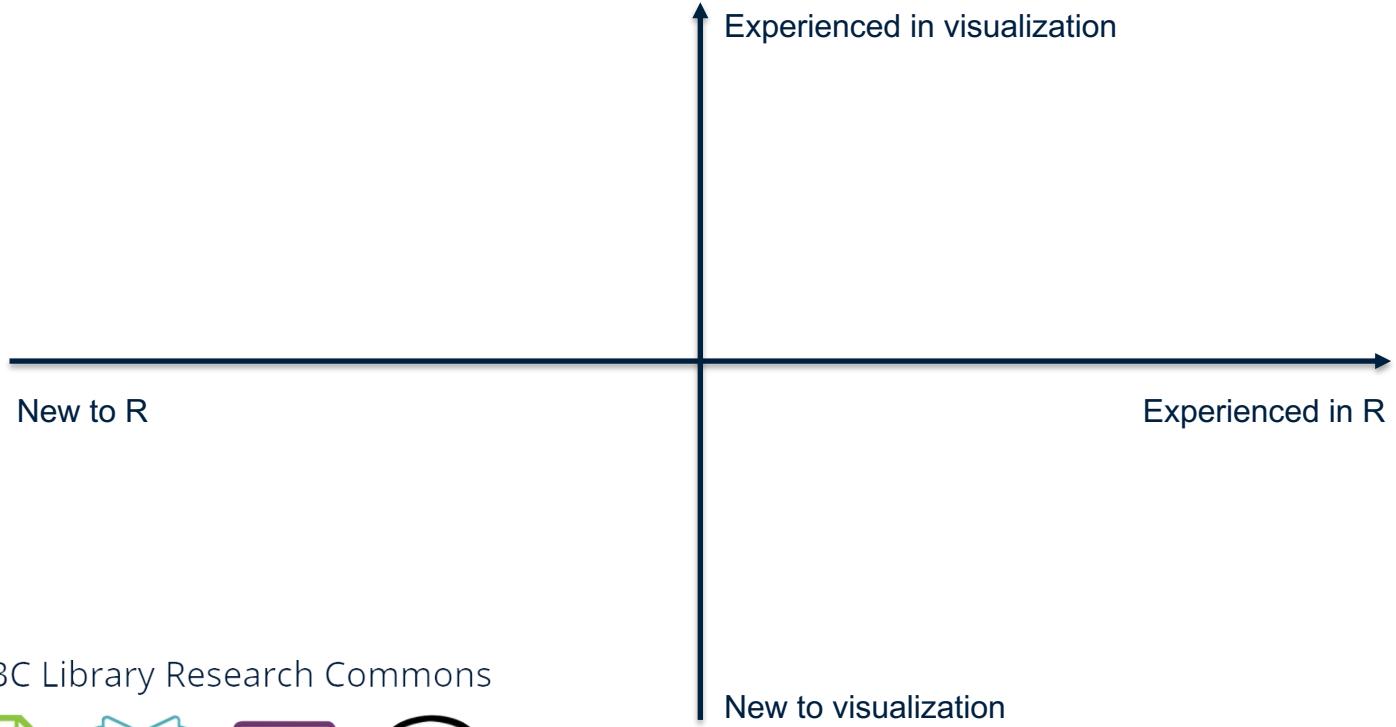
- Have **R** installed
- Have **R Studio** installed

Assumed prior knowledge:

- Have some experience developing visualizations or manipulating data in R (e.g. knowledge of ***ggplot2*** and ***dplyr***)

UBC Library Research Commons

YOUR EXPERIENCE WITH R AND VISUALIZATION



UBC Library Research Commons

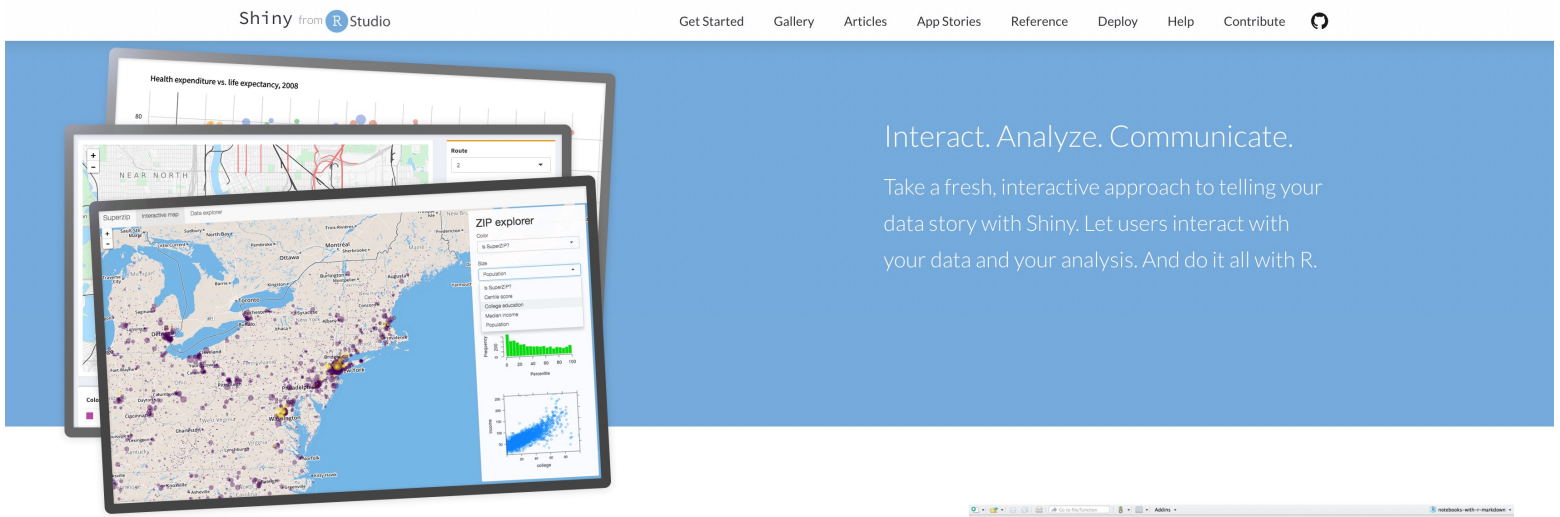
LEARNING OBJECTIVES

1. Become familiar with ***shiny*** package for making web applications
2. Learn how to make a simple shiny app in R
3. Learn how to expand on your app with more advanced features

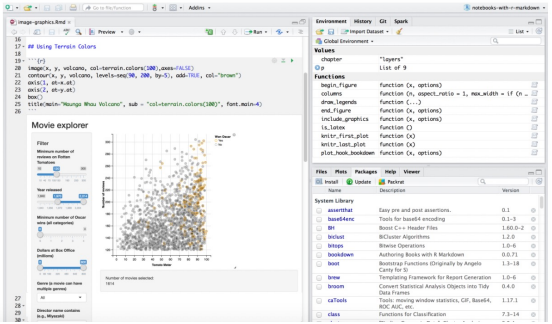
INTRODUCTION TO SHINY

UBC Library Research Commons

WHAT IS SHINY?



Shiny is an R package that makes it easy to build interactive web apps straight from R. You can host standalone apps on a webpage or embed them in R Markdown documents or build dashboards. You can also extend your Shiny apps with CSS themes, htmlwidgets, and JavaScript actions.



WHAT IS SHINY USED FOR?

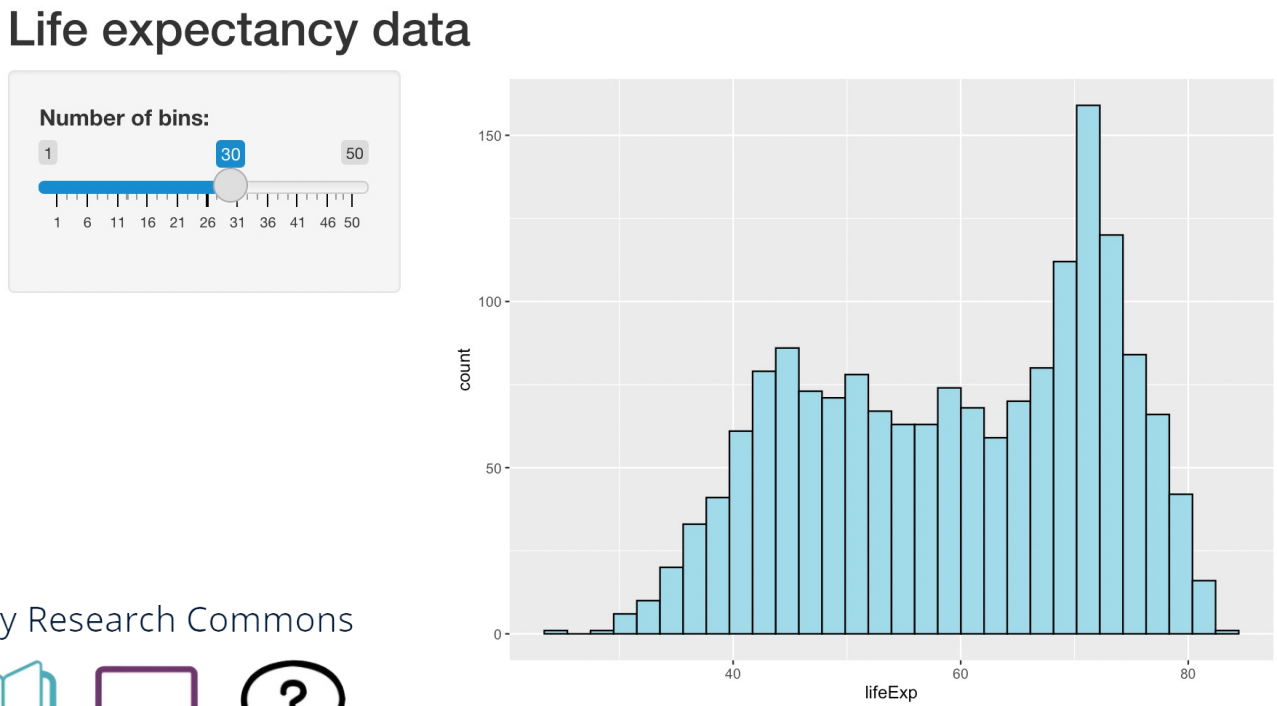
1. Developing interactive visualizations/tables (which can be incorporated on your website, in R markdown file, or in Tableau)
2. Developing data-oriented web applications
3. Developing full websites

STRUCTURE OF A SHINY APP

UI	Server	Launch app
Specifying user inputs and user outputs	Specifying how to construct outputs and do all calculations	Run *shinyApp(ui = ui, server = server)*

Note: previously you would have two separate files: ***UI.r*** and ***server.r***, now there is a combined ***app.R*** file

TIME TO SEE OUR FIRST SHINY APP!



UBC Library Research Commons

Shiny Widgets Gallery

For each widget below, the Current Value(s) window displays the value that the widget provides to shinyServer. Notice that the values change as you interact with the widgets.

Action button

Action

Current Value:

```
[1] 0
attr(,"class")
[1] "integer"                "shinyActionButtonValue"
```

See Code

Single checkbox

Choice A

Current Value:

```
[1] TRUE
```

See Code

Checkbox group

Choice 1
Choice 2
Choice 3

Current Values:

```
[1] "1"
```

See Code

Date input

2014-01-01

Current Value:

```
[1] "2014-01-01"
```

See Code

Date range

2022-03-26 to 2022-03-26

Current Values:

```
[1] "2022-03-26" "2022-03-26"
```

See Code

File input

Browse... No file selected

Current Value:

```
NULL
```

See Code

Numeric input

1

Current Value:

```
[1] 1
```

See Code

Radio buttons

Choice 1
Choice 2
Choice 3

Current Values:

```
[1] "1"
```

See Code

Select box

Choice 1

Current Value:

```
[1] "1"
```

See Code

Slider

0 50 100

0 10 20 30 40 50 60 70 80 90 100

Current Value:

```
[1] 50
```

Slider range

0 25 75 100

0 10 20 30 40 50 60 70 80 90 100

Current Values:

```
[1] 25 75
```

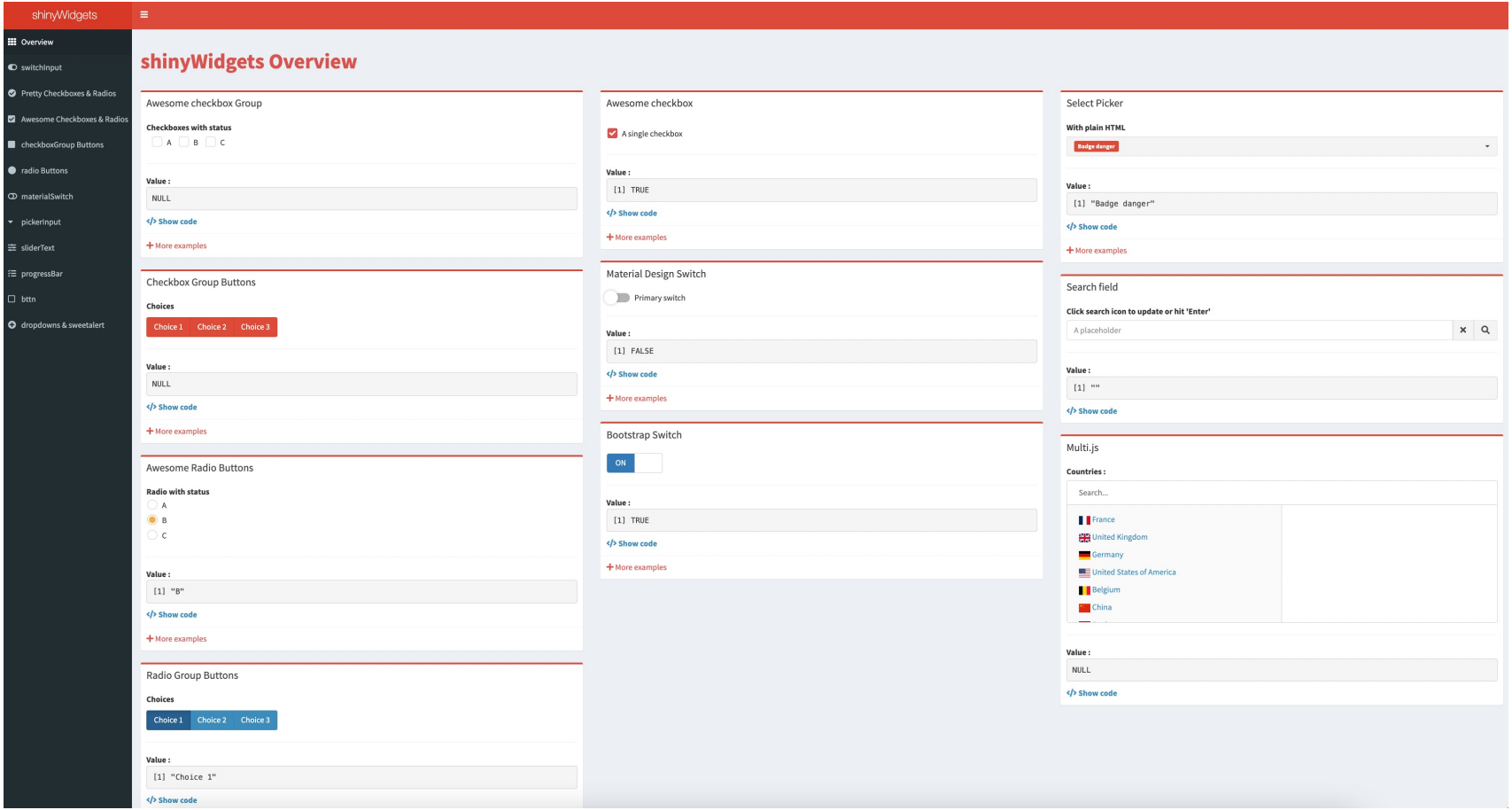
Text input

Enter text...

Current Value:

```
[1] "Enter text..."
```

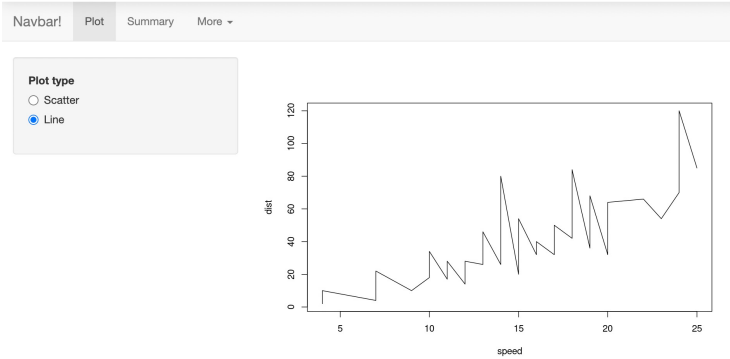
ADDITIONAL WIDGETS OFFERED BY *SHINYWIDGETS*



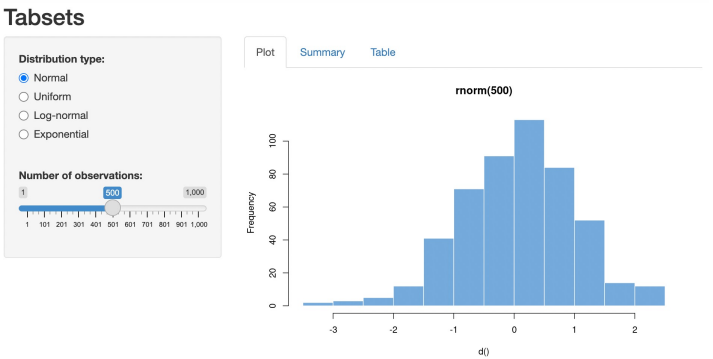
FOUR APPLICATION LAYOUTS

Horizontal/vertical layering

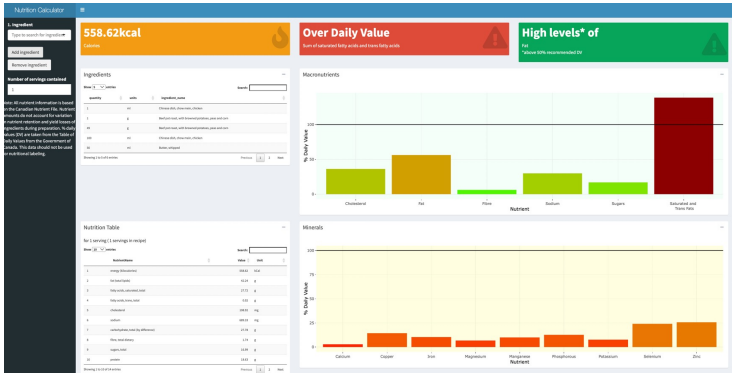
Navbar



Tabsets



ShinyDashboard



Gallery

Welcome to the Shiny Gallery! Below you can find a myriad of Shiny apps to be inspired by and to learn from. We have organized the apps in two main categories:

- **Shiny User Showcase** comprised of contributions from the Shiny app developer community.
- **Shiny Demos** that are designed to highlight specific features of shiny, the package.

Shiny User Showcase

The Shiny User Showcase is comprised of contributions from the Shiny app developer community. The apps are categorized into application areas and presented with a brief description, tags, and for many, the source code. Note that many of these apps are winners and honorable mentions of our annual Shiny contest!

Education

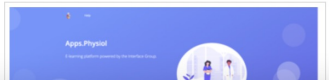
Apps designed for teaching

Didactic modeling process: Linear regression

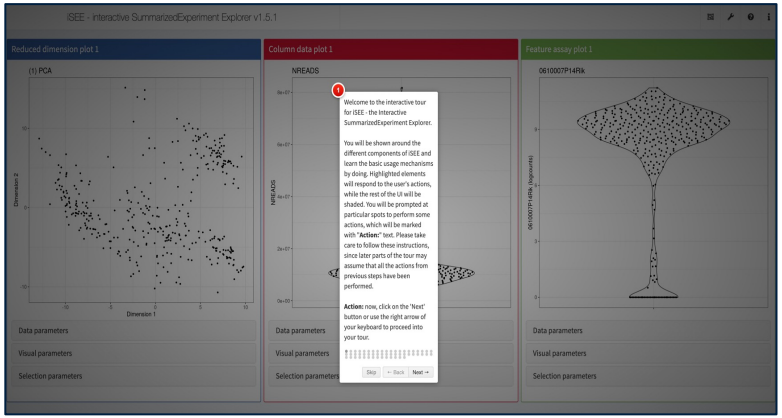
R/Shiny package to learn immune response modeling

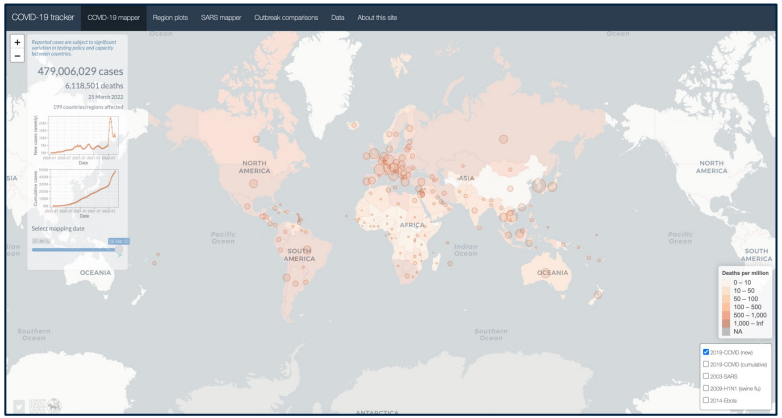
NCAA Swimming Team Finder for Incoming College Athletes

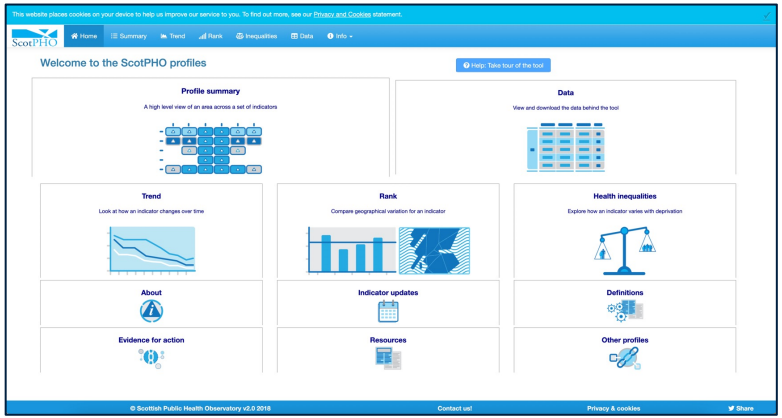
Radiant - A shiny app for statistics and machine learning



SHINY APP EXAMPLES





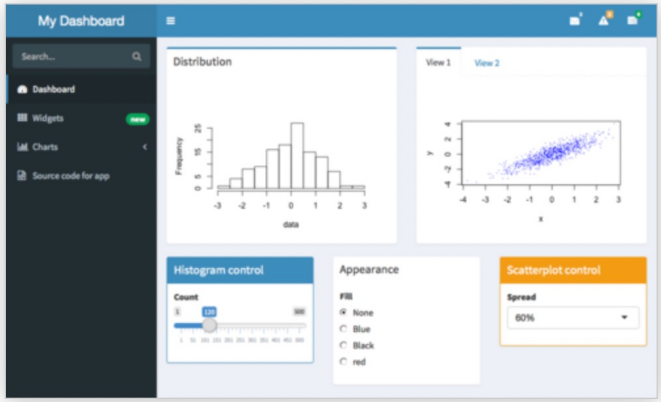


MODIFYING THE APP FURTHER

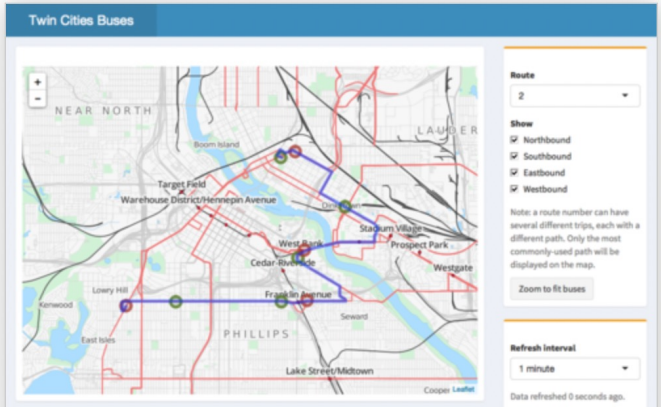
WITH

SHINY DASHBOARD

shinydashboard makes it easy to use Shiny to create dashboards like these:



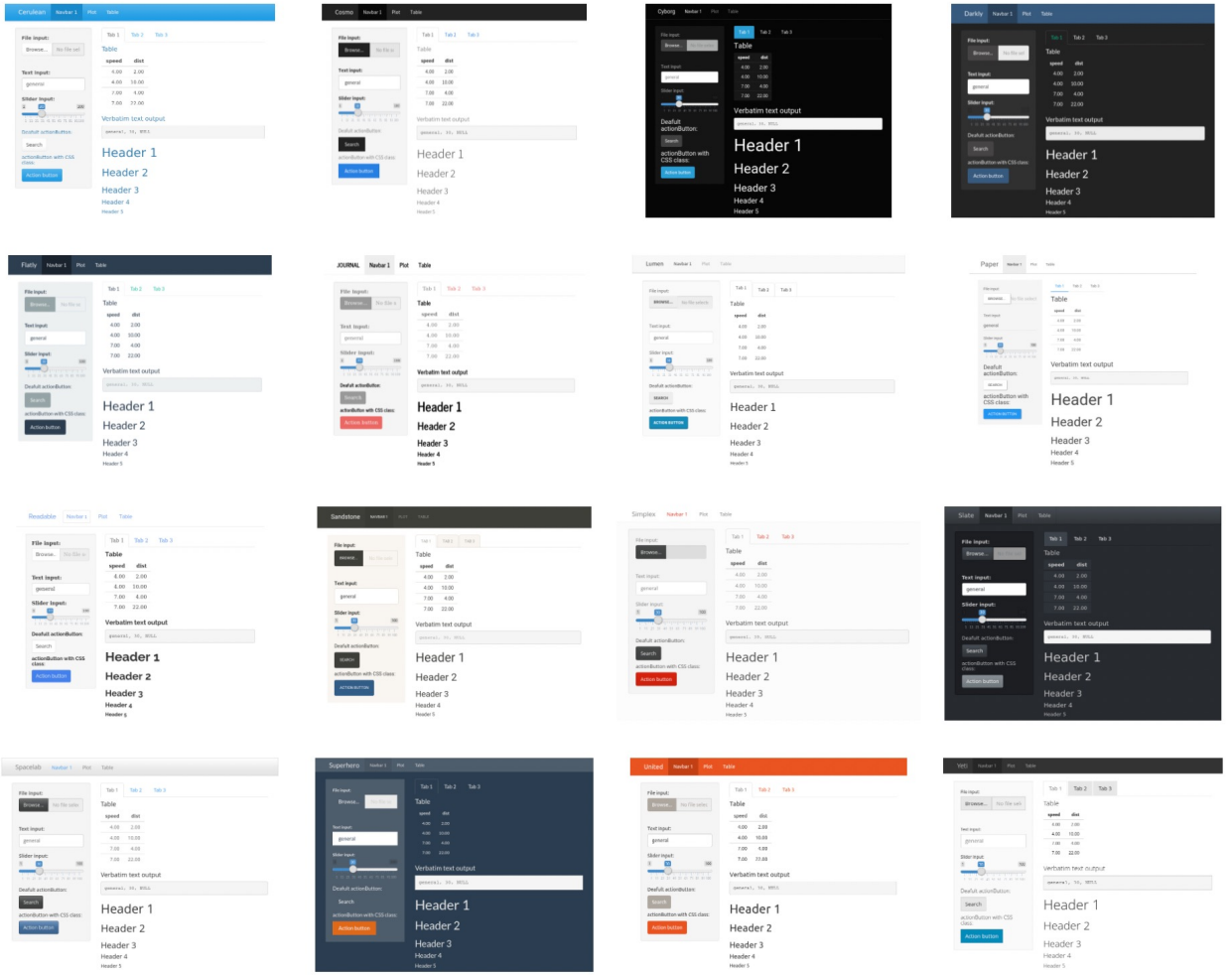
Dashboard



MODIFYING THE APP FURTHER

WITH

CSS THEMES



MODIFYING THE APP FURTHER

WITH

HTML WIDGETS

htmlwidgets for R

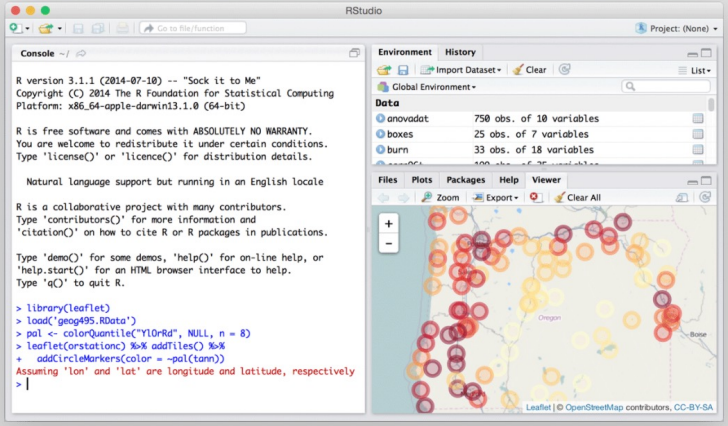
Home | Showcase | Develop | Flexdashboard | Crosstalk | Gallery | GitHub

Bring the best of JavaScript data visualization to R

Use JavaScript visualization libraries **at the R console**, just like plots

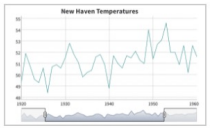
Embed widgets in **R Markdown** documents and **Shiny** web applications

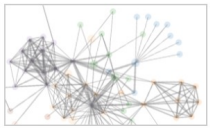
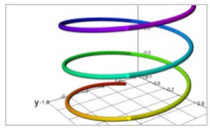
Develop new widgets using a framework that seamlessly bridges R and JavaScript



At the R console | In R Markdown docs | In Shiny apps

Widgets in action

Just a line or two of R code can be used to create interactive visualizations. See the featured widgets in the showcase and browse over 50 available widgets in the gallery.

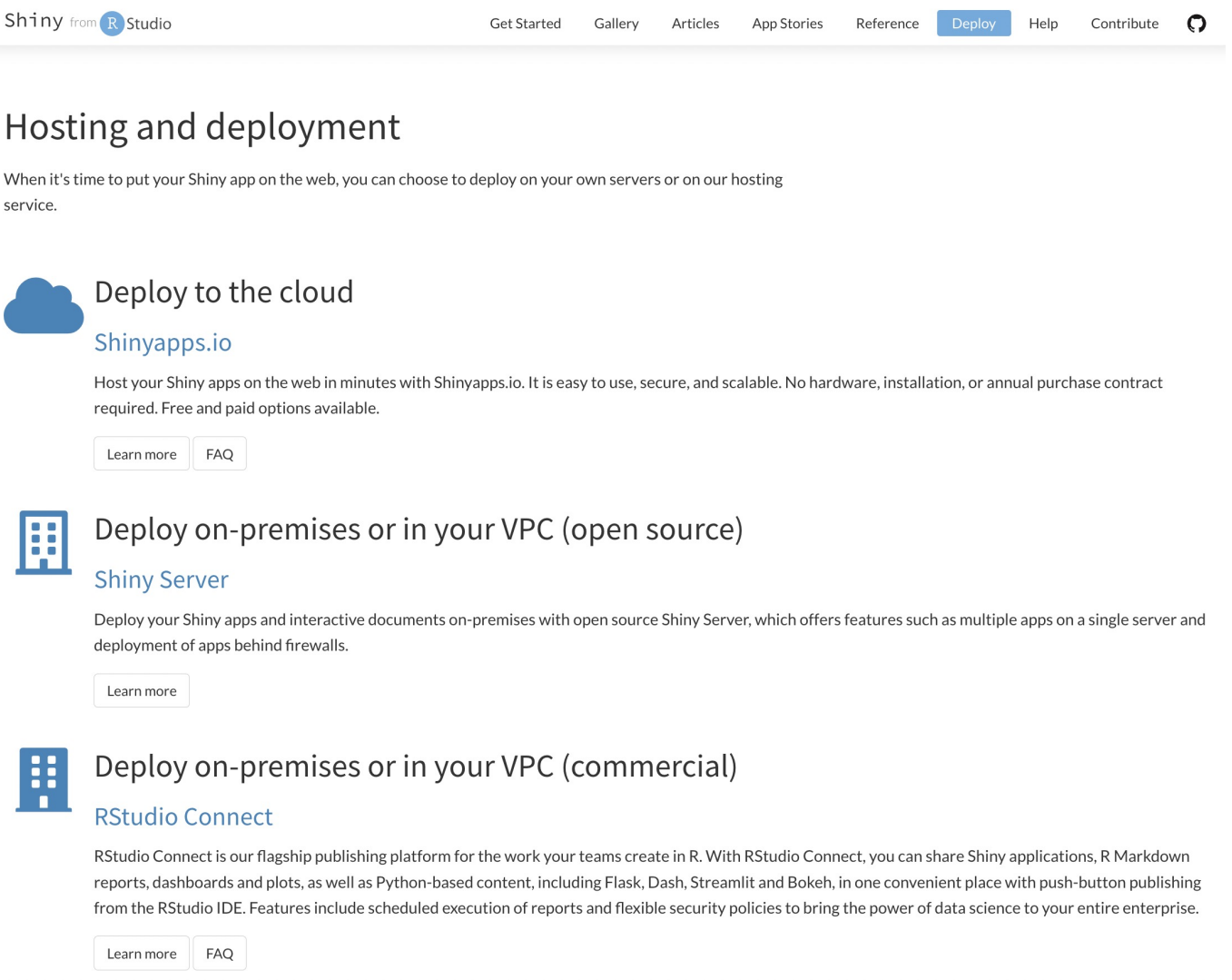
See the showcase »

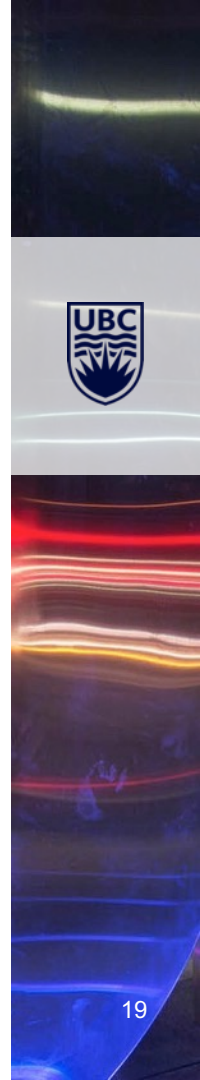
MODIFYING THE APP FURTHER

WITH

JAVA SCRIPT

THREE WAYS TO LAUNCH YOUR SHINY APP





DEPLOYING IT ON SHINYAPPS.IO

BENEFITS OF SHINY APPS

- Expansive interactivity
  - ✓ Great for data that should be presented interactively
- Breath and depth of content you can include
  - ✓ A project can have its own web page with multiple pages
- Diversity of features
  - ✓ From simple graphs and tables to networks and maps
- Works very well with many R packages
  - ✓ Shiny developers are constantly expanding shiny functionality
- Open-source and mostly free
- Lots of available educational resources, code examples you can use
  - ✓ Very active global community of R Shiny users and annual Shiny contests

“COSTS” OF SHINY APPS

- Requires a basic-intermediate understanding of coding
  - ✓ Significant time commitment
- Restrictions on publishing the app and using it for free
  - E.g. 25 free user hours/month
- Getting to 90% of what you want is easy, the other 10% can take forever
- Sadly doesn’t work with all R packages
- Things can take a long time to load and process
  - Depending on the size of your data and what you are visualizing

WHAT DO YOU NEED TO KNOW TO BUILD SHINY APPS?

MUST know:

- Making basic and static visualizations in R (e.g. knowledge *ggplot2* package)
- Cleaning/manipulating your data in R (e.g. knowledge of *dplyr, tidyr* packages)
- Writing out simple Shiny syntax

OPTIONAL but USEFUL to know:

- Making interactive visualizations in R (e.g. *plotly, chorddiag, visNetwork, tmap* packages)
- Knowing some *html* and *css* to modify various visual aspects of the app
- Some prior experience in non-R coding can be helpful (but not necessary)

LET'S MAKE OUR FIRST SHINY APP!

Gapminder Shiny App

Country

Afghanistan, Albania, Algeria, Angola, Argenti

Continent

☑ Africa ☑ Americas ☑ Asia ☐ Europe
☐ Oceania

Life Expectancy

20 50 80

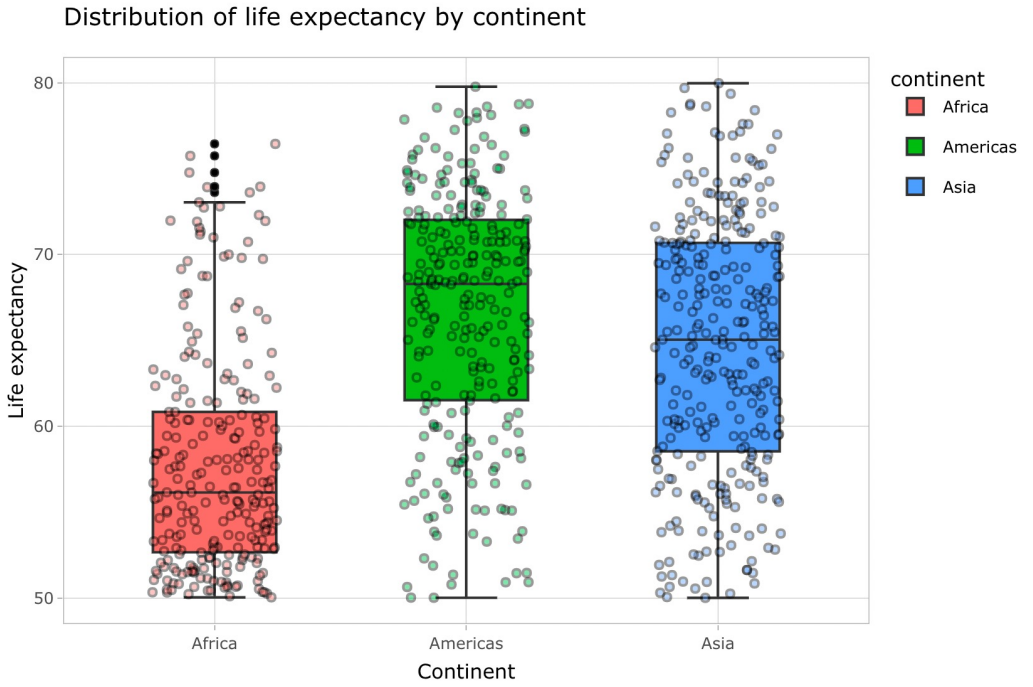
20 27 34 41 48 55 62 69 76 83

Year

1952, 1957, 1962, 1967, 1972, 1977, 1982, 1

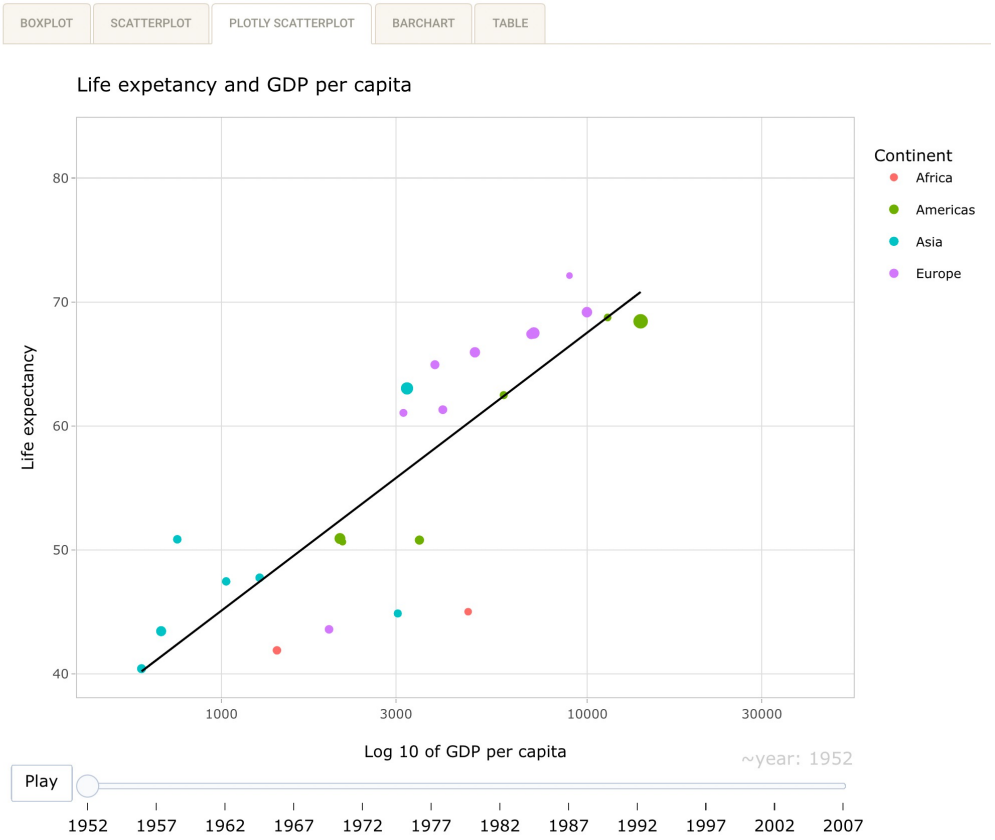
Reset all inputs

Boxplot Table

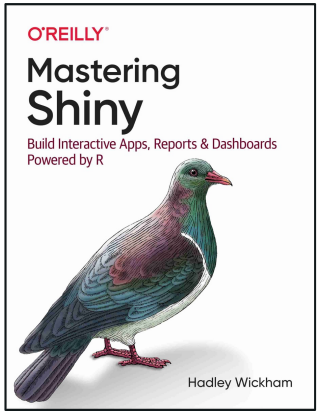


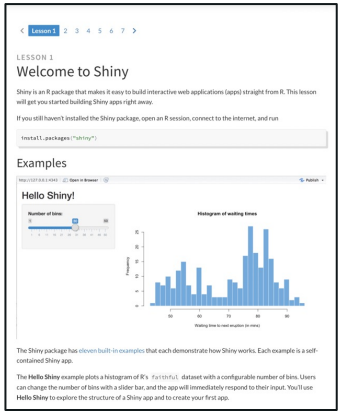
ADDING MORE FEATURES TO OUR SHINY APP

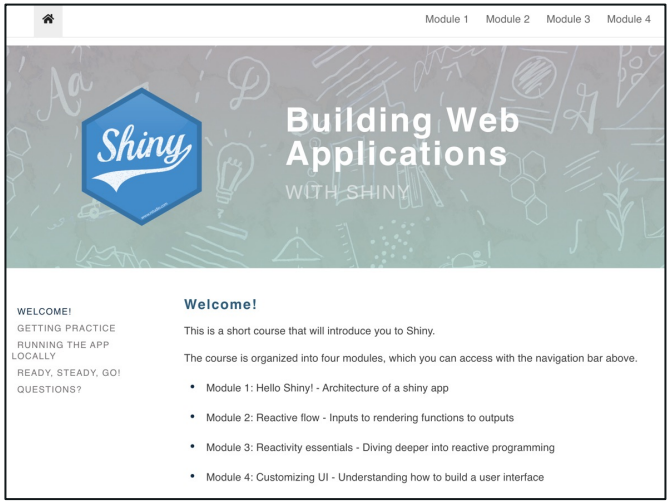
Gapminder Shiny App



SHINY RESOURCES





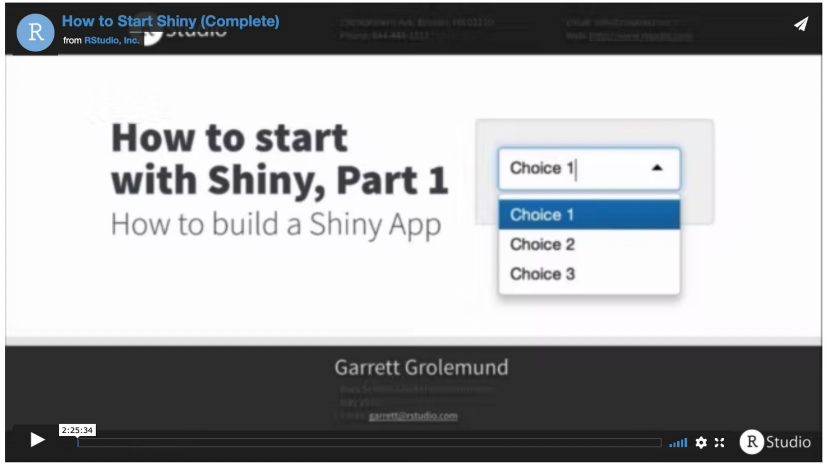


Shiny from Studio

Get Started | Gallery | Articles | App Stories | Reference | Deploy | Help | Contribute

How to Start Shiny tutorial

The How to Start Shiny video series will take you from R programmer to Shiny developer. Watch the complete tutorial, or jump to a specific chapter by clicking a link belo
The entire tutorial is two hours and 25 minutes long. Download the slides and exercises here.



Part 1 - How to build a Shiny app

1. Introduction
2. R
3. App architecture
4. App template
5. Inputs and outputs
6. The server function
7. Sharing apps
8. Shinyapps.io
9. Shiny servers
10. Recap - Part 1

Part 2 - How to customize reactions

11. Introduction
12. Review of Part 1
13. Reactivity
14. Reactive values
15. Reactive functions
16. render*()
17. reactive()
18. isolate()
19. observeEvent()
20. eventReactive()
21. reactiveValues()
22. Recap - Part 2
23. Parting tips

Part 3 - How to customize appearance

24. Introduction
25. Review of Parts 1 and 2
26. HTML UI
27. Adding static content
28. Building layouts
29. Panels and tabsets
30. Prepackaged layouts
31. CSS
32. Recap - Part 3

THANK YOU FOR ATTENDING!

Please fill out a short feedback survey 😀

UBC Library Research Commons