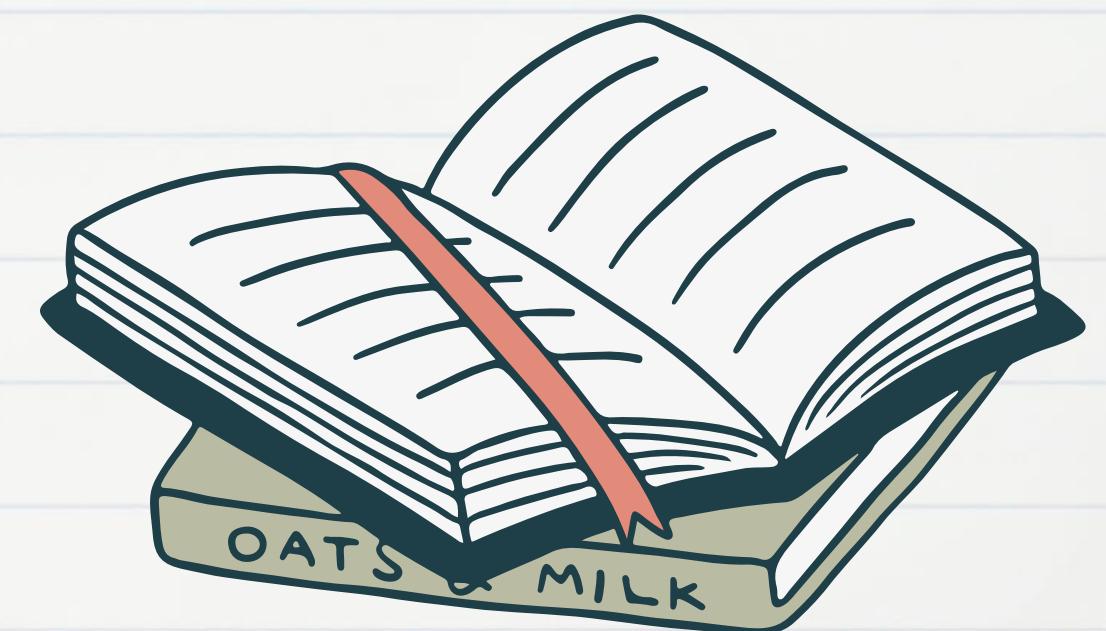


From Data to Actions with **R Shiny**

Presented by: Lillian Lu

Overview

- Introduction
- Shiny use case
- Shiny components
- 20 mins break
- Workshop
- Resources
- Q & A



Introduction



- Data scientist/analyst/consultant
- Master's degree in Analytics
- Originally from China
- Bachelor's degree in Design
- Moved to New Zealand in 2012
- 10 years working as a digital marketer

Introduction



- Data scientist/analyst/consultant
- Master's degree in Analytics
- Originally from China
- Bachelor's degree in Design
- Moved to New Zealand in 2012
- 10 years working as a digital marketer

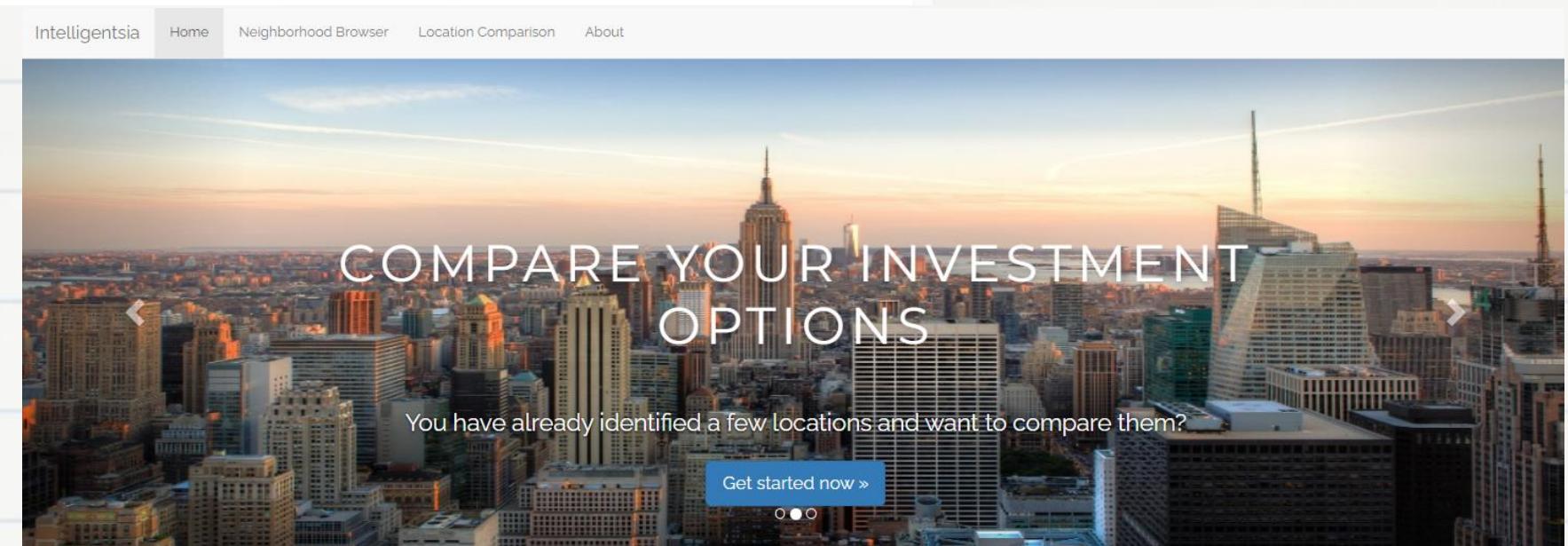
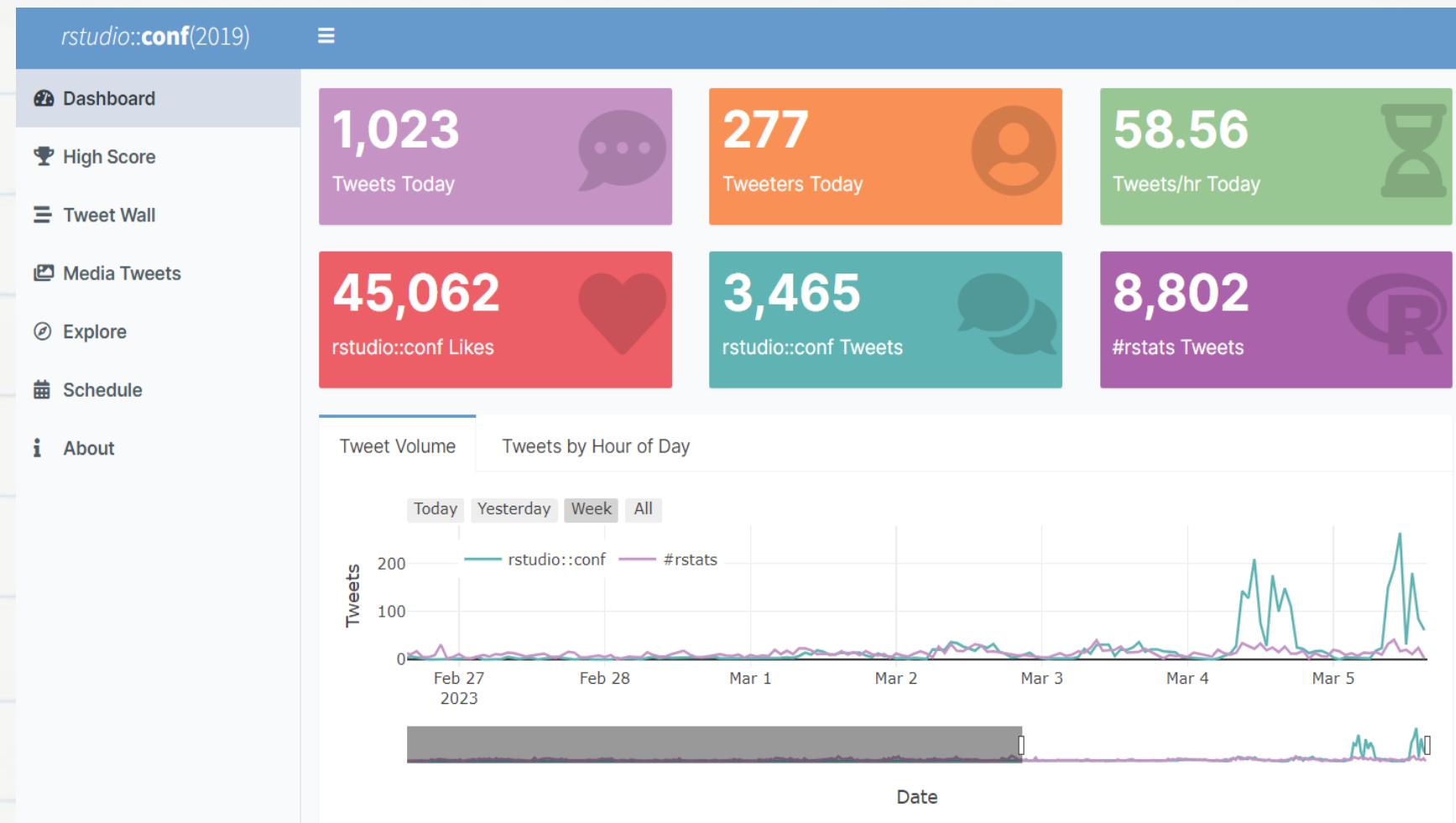
WellingtonNZ



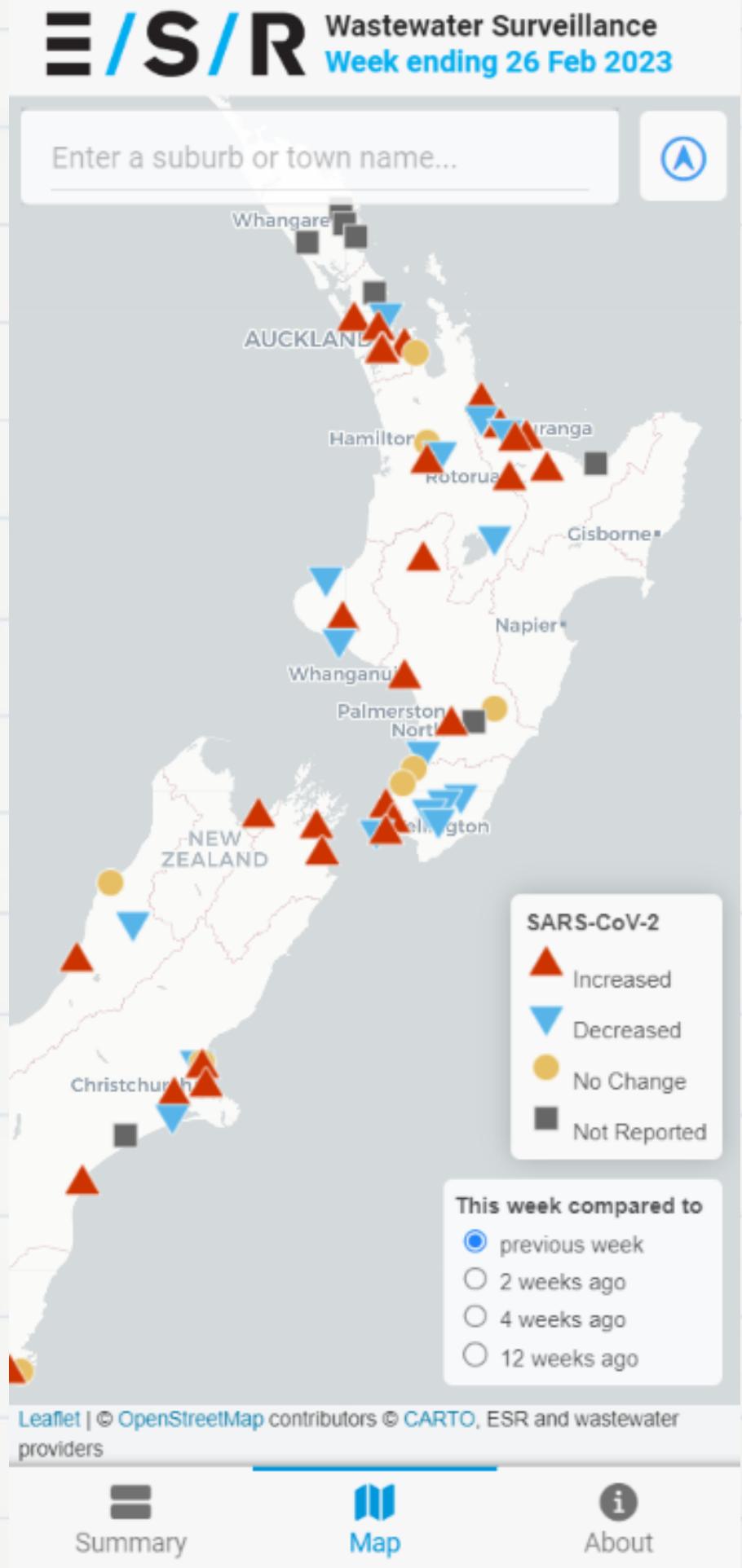
What's R shiny?



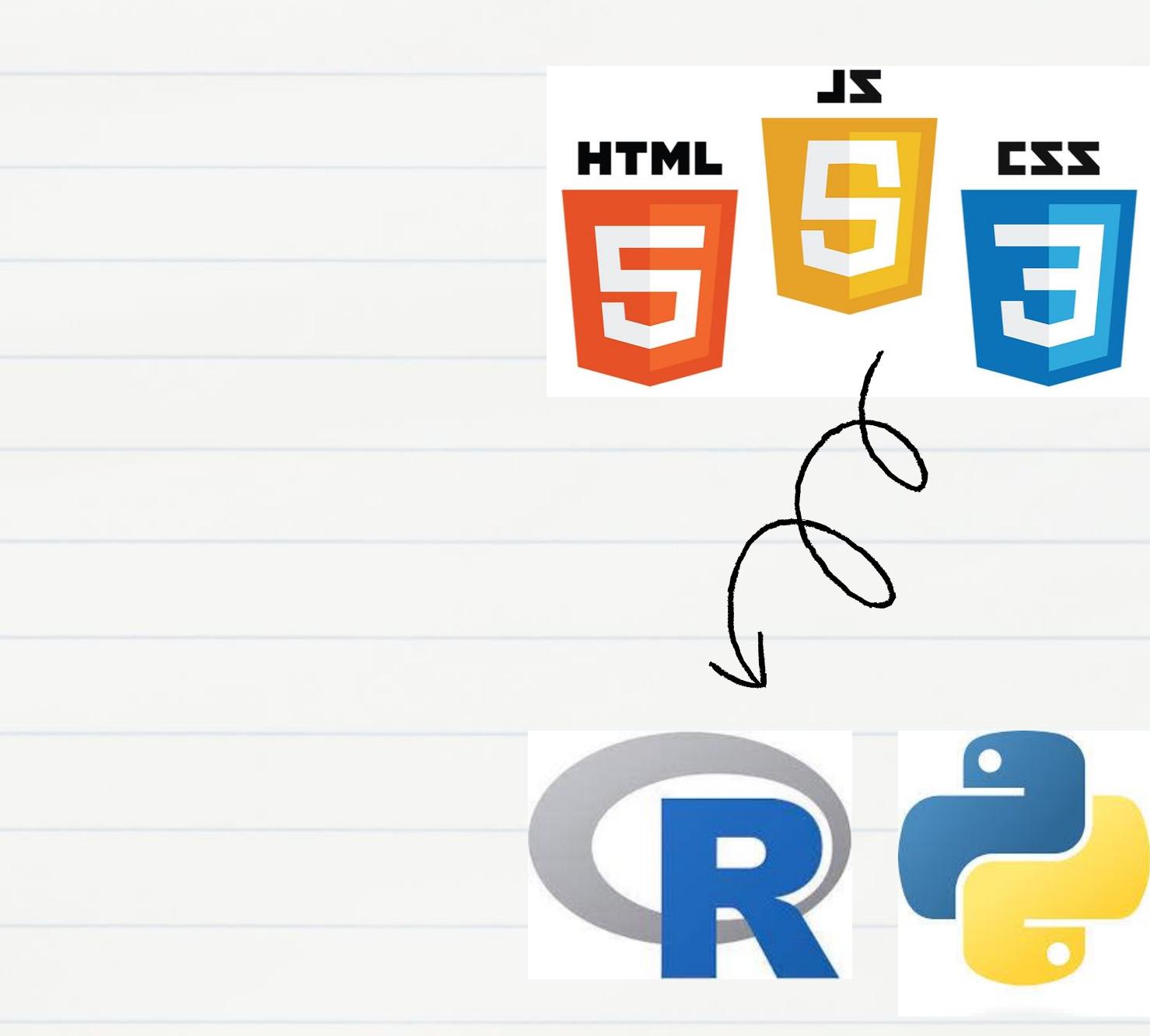
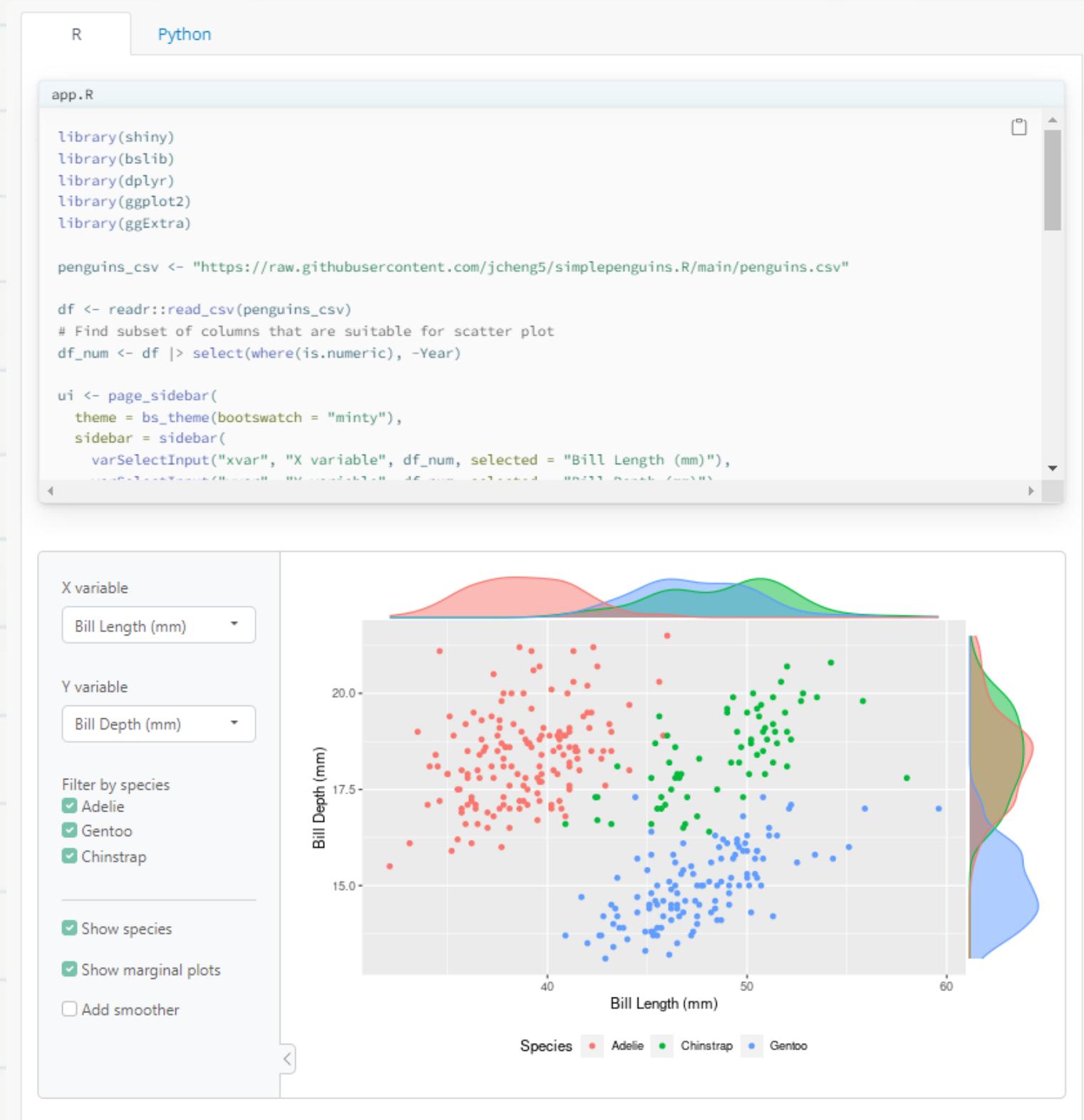
A screenshot of a web browser window showing the Shiny homepage. The page has a dark background with a large white header containing the Shiny logo and the text "Easy web apps for data science without the compromises". Below the header is a sub-header "No web development skills required". At the bottom of the page is a horizontal bar with two buttons: "Get started in R" and "Get started in Python". The background of the browser window shows a blurred view of other tabs and the operating system's taskbar.



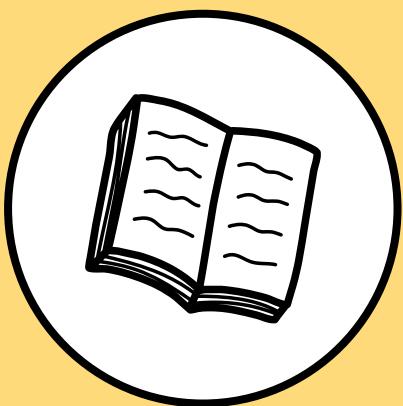
WE GOT YOU COVERED!



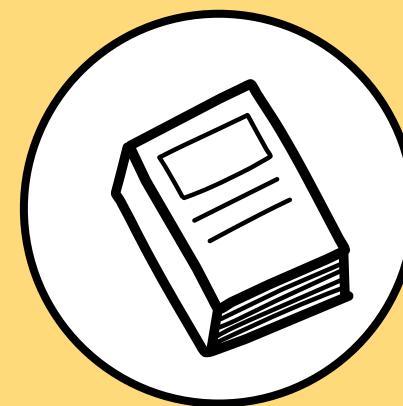
What's R shiny?



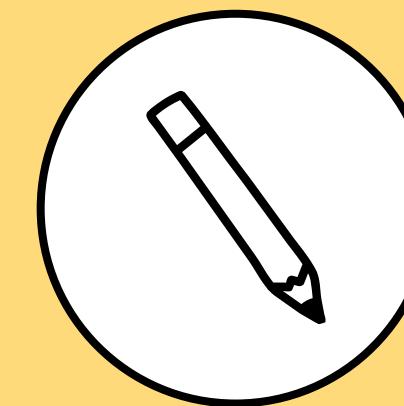
Shiny use case



Prototype



Work showcase



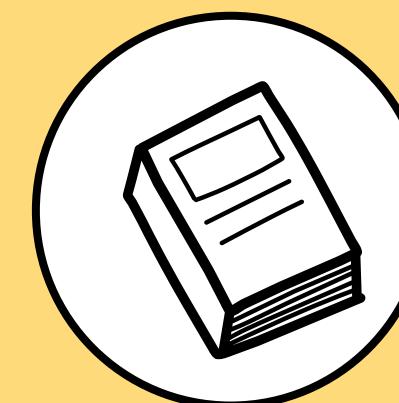
Data tool



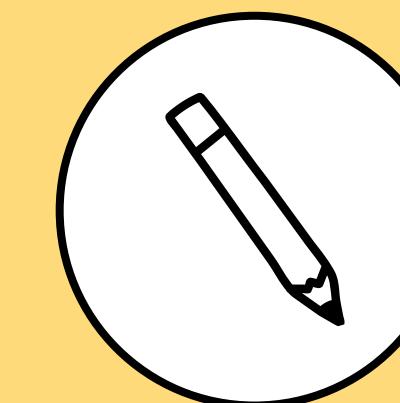
Shiny use case



Prototype



Work showcase



Data tool



BI Dashboard



Shiny use case



Prototype

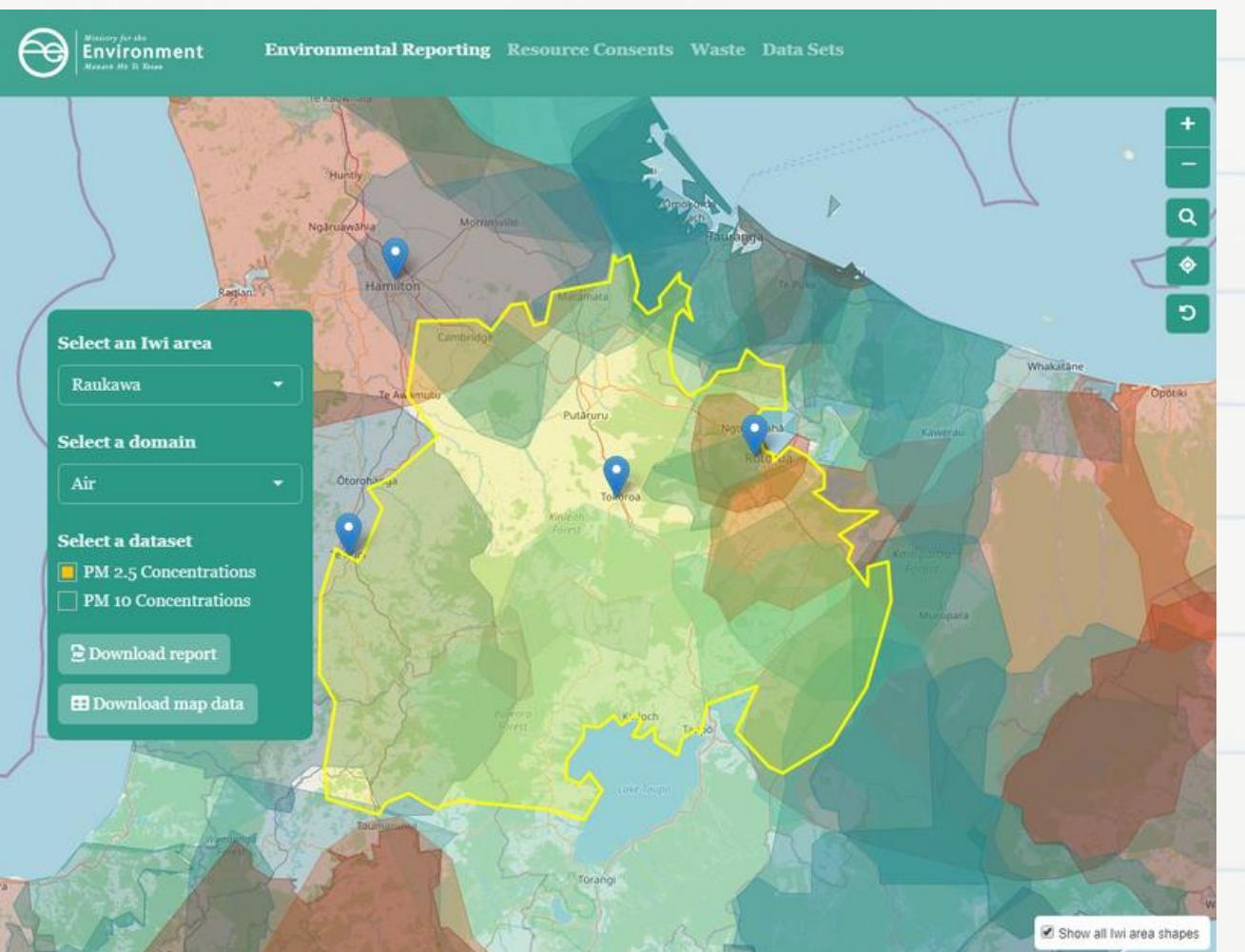
Create interactive prototypes
that allow you to effectively
present and sell your ideas to
stakeholders.

Shiny use case

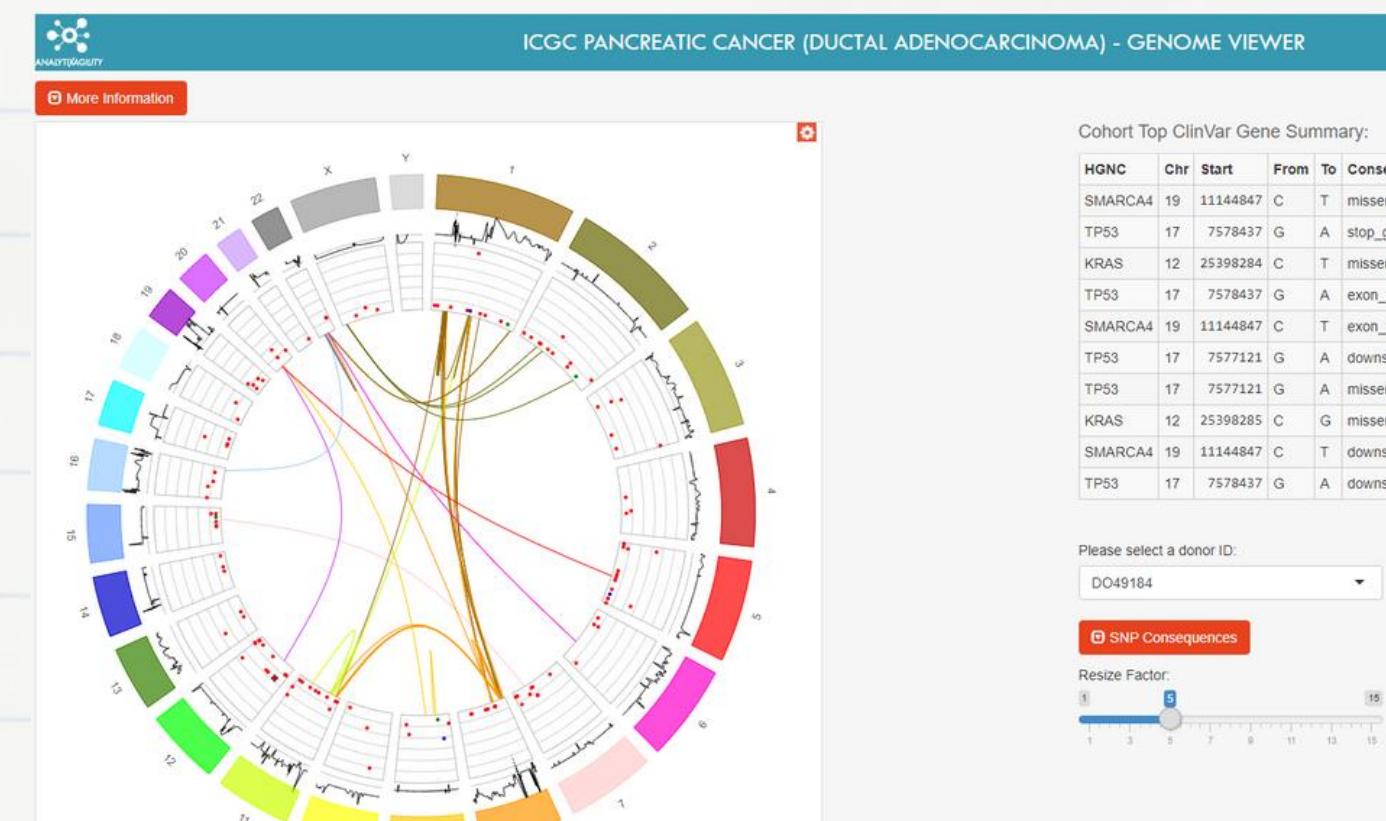


Prototype

Create interactive prototypes
that allow you to effectively
present and sell your ideas to
stakeholders.



Environmental data by iwi boundaries



Genome browser

Shiny use case



Work Showcase

Showcase machine learning
models and research findings
interactively for engaging
communication.

Shiny use case



Work Showcase

Showcase machine learning models and research findings interactively for engaging communication.

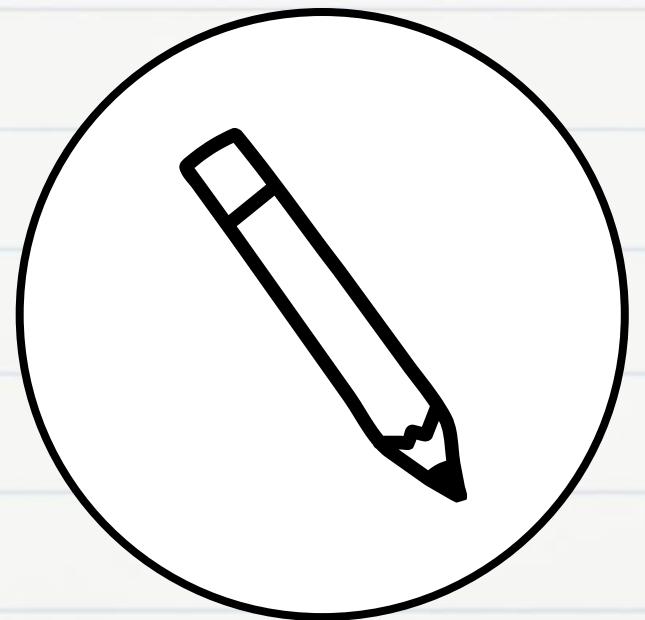
The screenshot shows a modal window titled "COVID-19 Vaccine Strategies". The content includes an "Introduction" section with text about a paper and a "Next" button. Below the introduction is a thumbnail image of a woman smiling, with the caption "THE LANCET Regional Health Western Pacific". At the bottom of the modal are "Back" and "Next" buttons, along with a progress bar and the number "0.2". The background of the main dashboard shows a dark grey header with the E/S/R logo and the title "COVID-19 Vaccination M".

COVID-19 Vaccination Modelling

The screenshot shows a dark-themed application titled "Shiny MRI". It features a dropdown menu "Choose a Demo Data" set to "3D Brain MRI", and a horizontal slider with a value of "69" ranging from 1 to 176. Below the slider is a grayscale 3D MRI reconstruction of a human brain. The top navigation bar includes "Shiny MRI", "Home", and "About". On the right side, there is a faint text "Or" followed by a link "View other MRI data".

ShinyMRI - View MRI images in Shiny

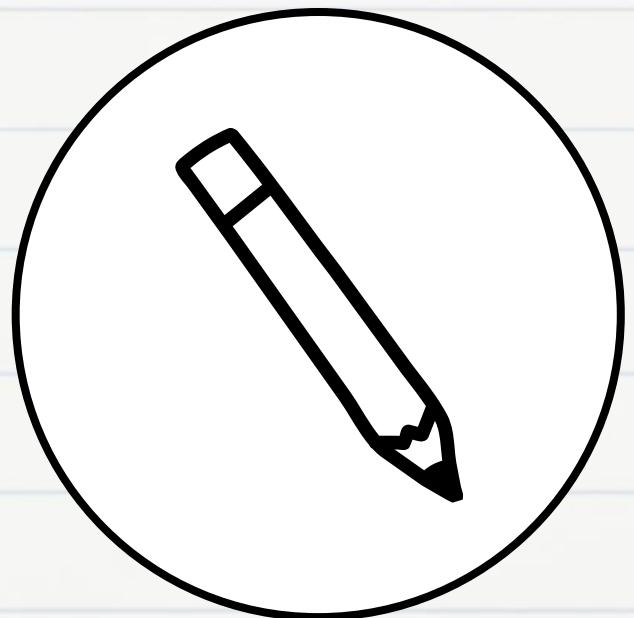
Shiny use case



Data tool

Shiny empowers you to become a self-sufficient, full-stack data scientist - no more waiting on service tickets!

Shiny use case



Data tool

Shiny empowers you to become a self-sufficient, full-stack data scientist - no more waiting on service tickets!

Didactic modeling process: Linear regression for a safety

Let's get started Step 1 Step 2 Step 3 Step 4 Step 5

Through this application, it is intended to develop a learning environment method of ordinary least squares. In any case, we will focus on the process worry! you will find alternatives to learn all these technical aspects independently

The data used in this application are publicly available on the page of this public entity correspond to a series of social, educational, sports and cultural activities.

Show entries

Subregion	Municipality	Projected population	Thefts	Traffic
All	All	All	All	All
Valle de Aburra	Barbosa	50.836	115	
Valle de Aburra	Caldas	78.756	141	

Didactic modeling process: Linear regression



NMS project management app

Shiny use case



BI Dashboard

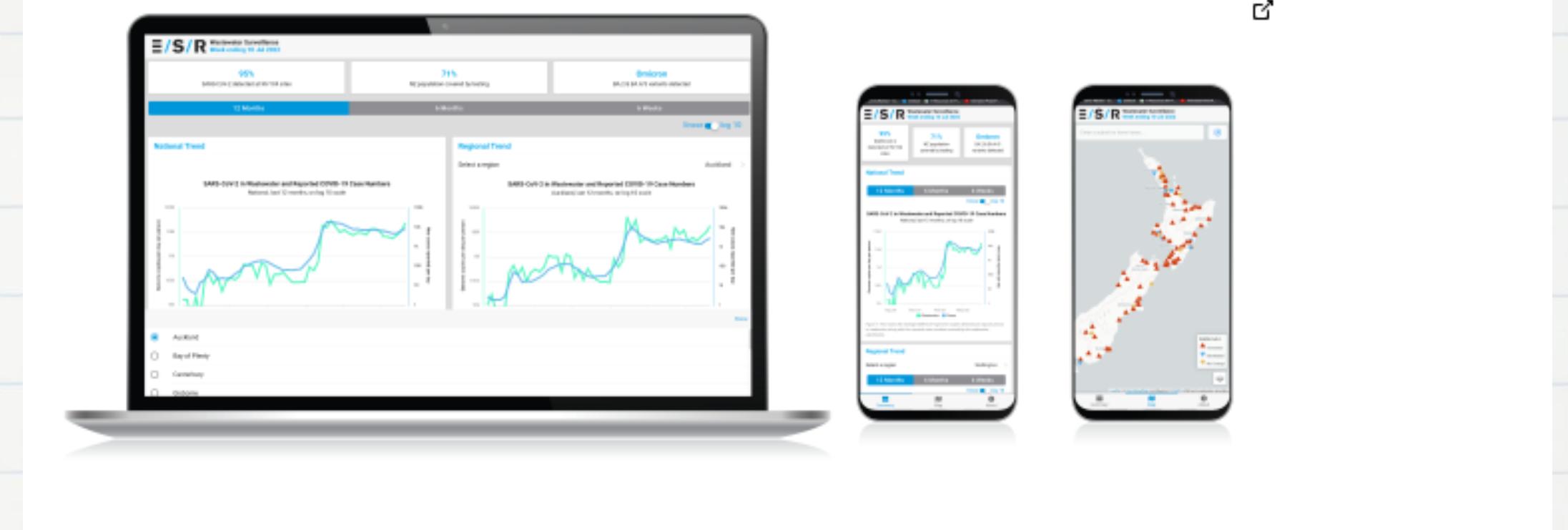
- Steep learning curve hence harder to maintain
- Complexity for simple reports
- Lack of native BI features such as drag-and-drop or user access controls
- Scaling Shiny apps for enterprise-level usage may require additional effort and resources

case study



Wastewater Surveillance Dashboard

This dashboard is designed to share ESR's wastewater science, and help the public track potential COVID-19 risks in their local areas with easy-to-digest data visualisation. It is optimised for both desktop and mobile use.



Shiny

Shiny components

```
library(shiny)  
ui <- fluidPage()  
  
server <- function(input, output, session) {}  
  
shinyApp(ui = ui, server = server)
```

User interface

controls the layout and appearance of app

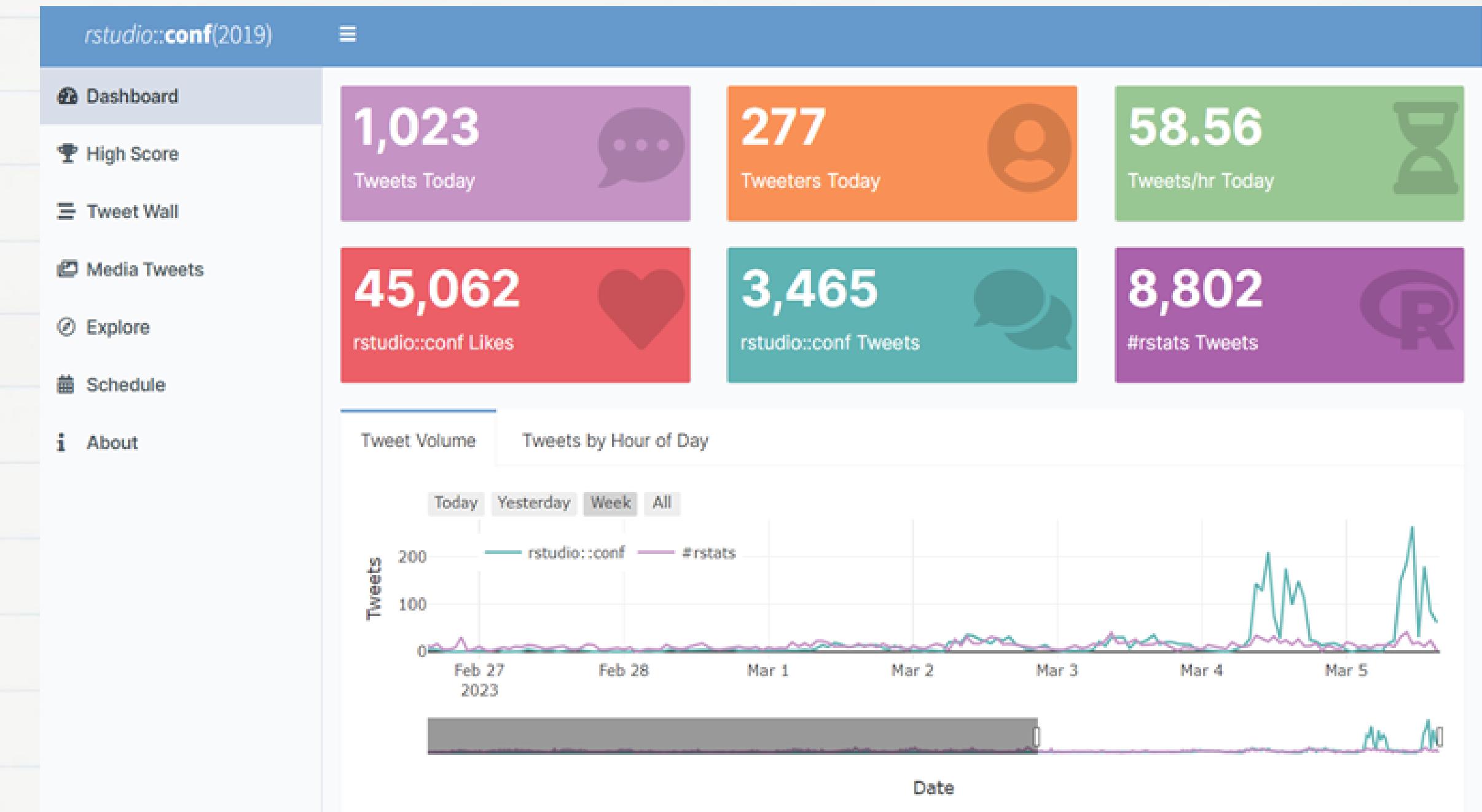
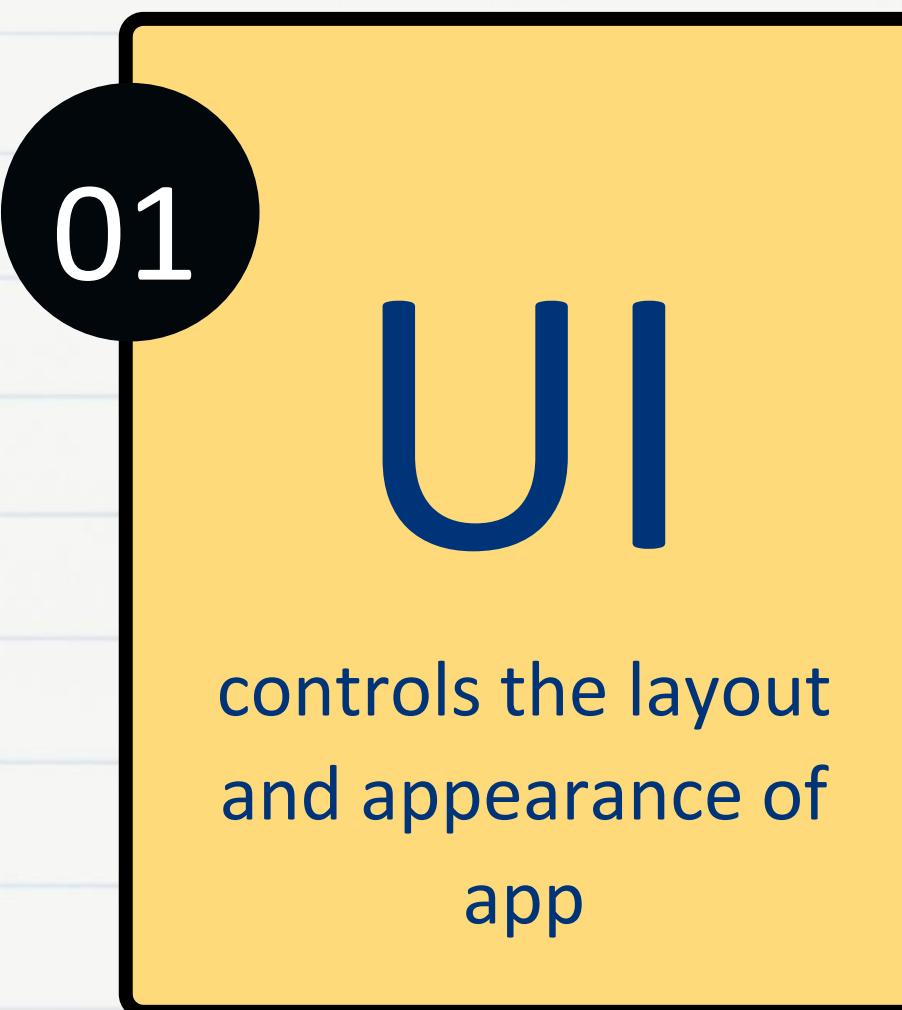
Server function

contains instructions needed to build app

shinyApp()

Creates the Shiny app object

Shiny components



Shiny components

01

UI

controls the layout
and appearance of
app

Inputs

collect values from the user

Access the current value of an input object with **input\$<inputId>**. Input values are **reactive**.

Action

actionButton(inputId, label, icon, ...)

Link

- Choice 1
- Choice 2
- Choice 3
- Check me



actionLink(inputId, label, icon, ...)

checkboxGroupInput(inputId, label, choices, selected, inline)

checkboxInput(inputId, label, value)

dateInput(inputId, label, value, min, max, format, startview, weekstart, language)

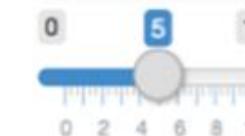
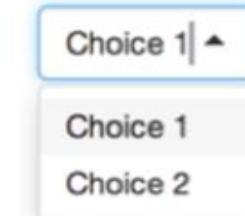
dateRangeInput(inputId, label, start, end, min, max, format, startview, weekstart, language, separator)

Choose File

1

.....

- Choice A
- Choice B
- Choice C



Apply Changes

Enter text

fileInput(inputId, label, multiple, accept)

numericInput(inputId, label, value, min, max, step)

passwordInput(inputId, label, value)

radioButtons(inputId, label, choices, selected, inline)

selectInput(inputId, label, choices, selected, multiple, selectize, width, size) (also `selectizeInput()`)

sliderInput(inputId, label, min, max, value, step, round, format, locale, ticks, animate, width, sep, pre, post)

submitButton(text, icon)
(Prevents reactions across entire app)

textInput(inputId, label, value)

Shiny components

02

SERVER

contains instructions
needed to build app

When ... Do ...

Shiny components

02

SERVER

contains instructions
needed to build app

When... Do...

When the button
is clicked

Do generate a plot

Shiny components

02

SERVER

contains instructions
needed to build app

When... Do...

When the button
is clicked **Do** generate a
plot

When select a suburb **Do** zoom into
the area

Shiny components

02

SERVER

contains instructions
needed to build app

The diagram illustrates a Scratch script structure. It features two main horizontal rows. The top row has the word "When" in large blue letters on the left, followed by three blue dots, a vertical yellow dashed line, and the word "Do" in large blue letters on the right, followed by three blue dots. The bottom row has the word "When" in large blue letters on the left, followed by a vertical yellow dashed line and the word "select a suburb" in green text, and "Do" in large blue letters on the right, followed by a vertical yellow dashed line and the text "zoom into the area" in green.

When ... Do ...

When the button is clicked Do generate a plot

When select a suburb Do zoom into the area

Shiny components

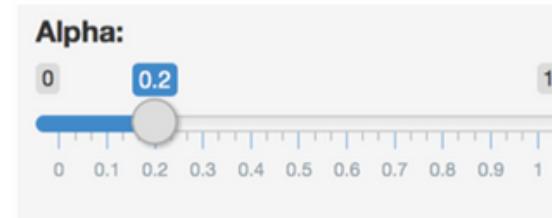
02

SERVER

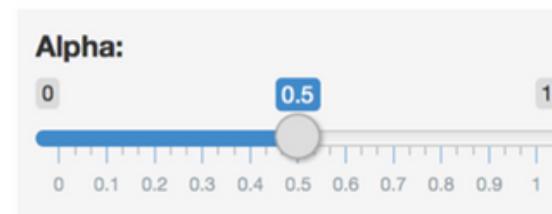
contains instructions
needed to build app

```
# Set alpha level
sliderInput(inputId = "alpha",
            label = "Alpha:",
            min = 0, max = 1,
            value = 0.5)
```

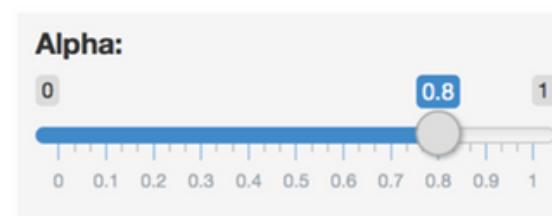
input\$alpha



input\$alpha = 0.2



input\$alpha = 0.5



input\$alpha = 0.8

input\$x

Create your
own reactive values
*Input()

expression()

Update

output\$y

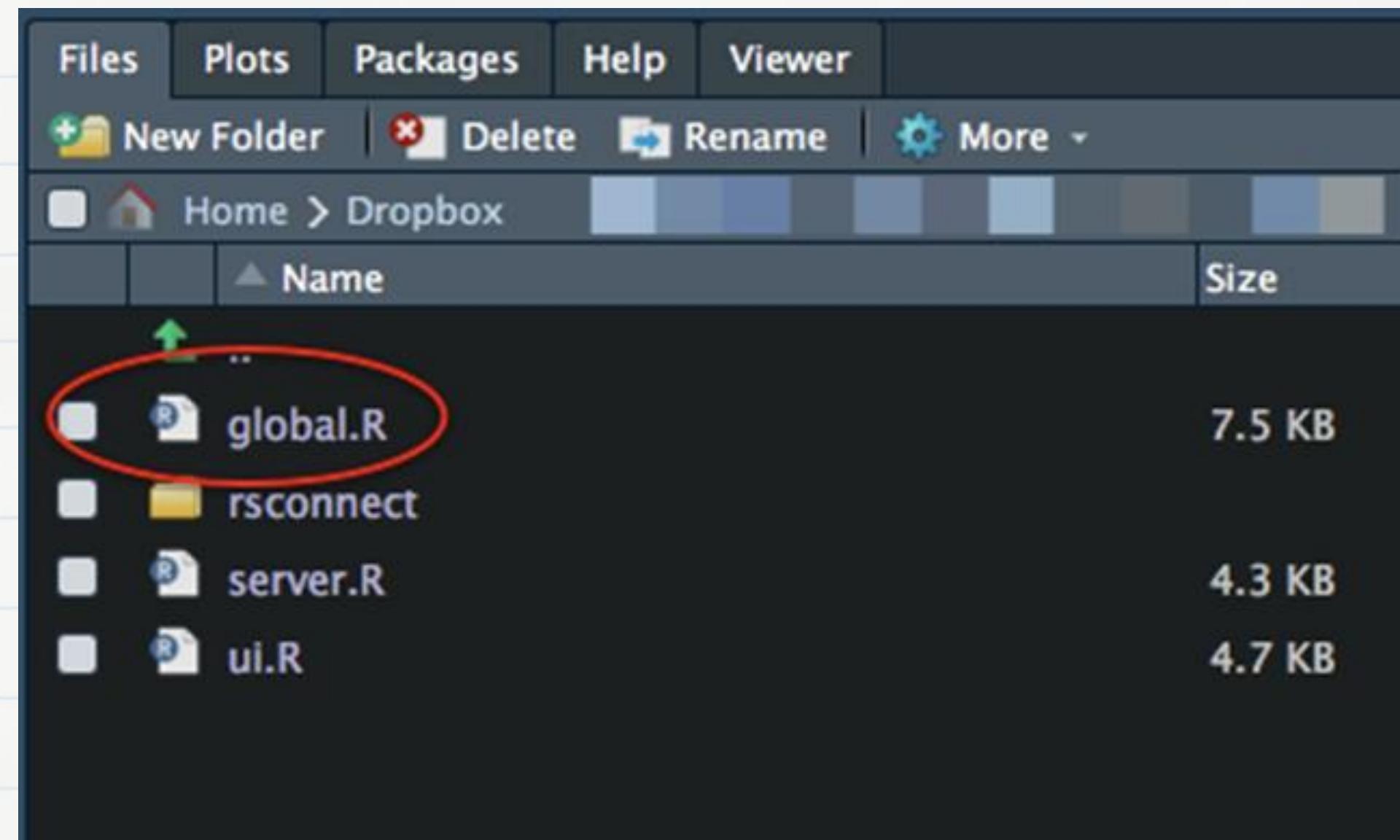
Render
reactive output
render*()

Shiny components

03

GLOBAL

A script to load
libraries, source
functions, clean data
& more.





Break

Workshop prerequisite

1. Have R Studio installed locally or use the cloud version:

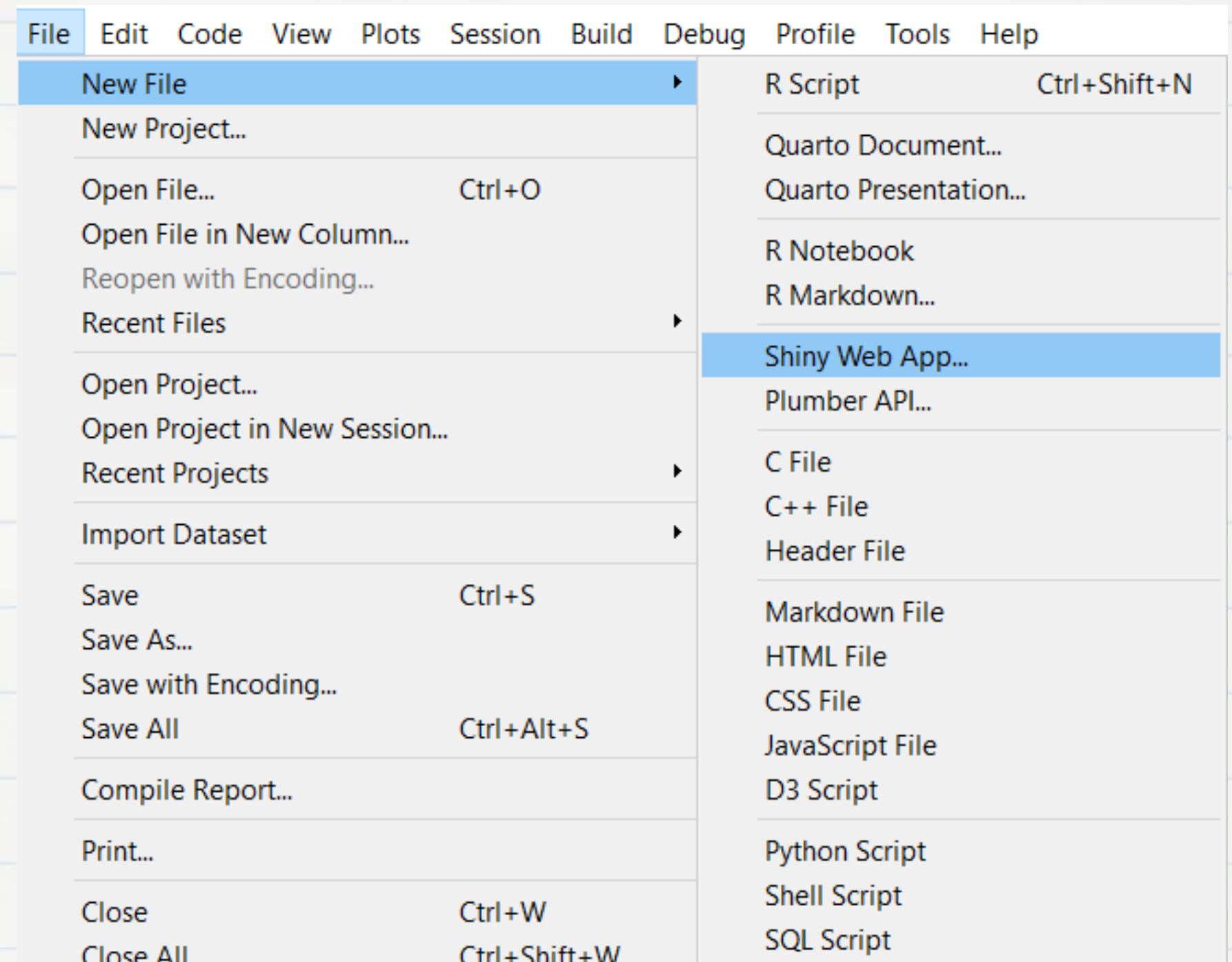
posit.cloud



3. Download the files from
github.com/lillianlu-nz/ShinyWorkshop

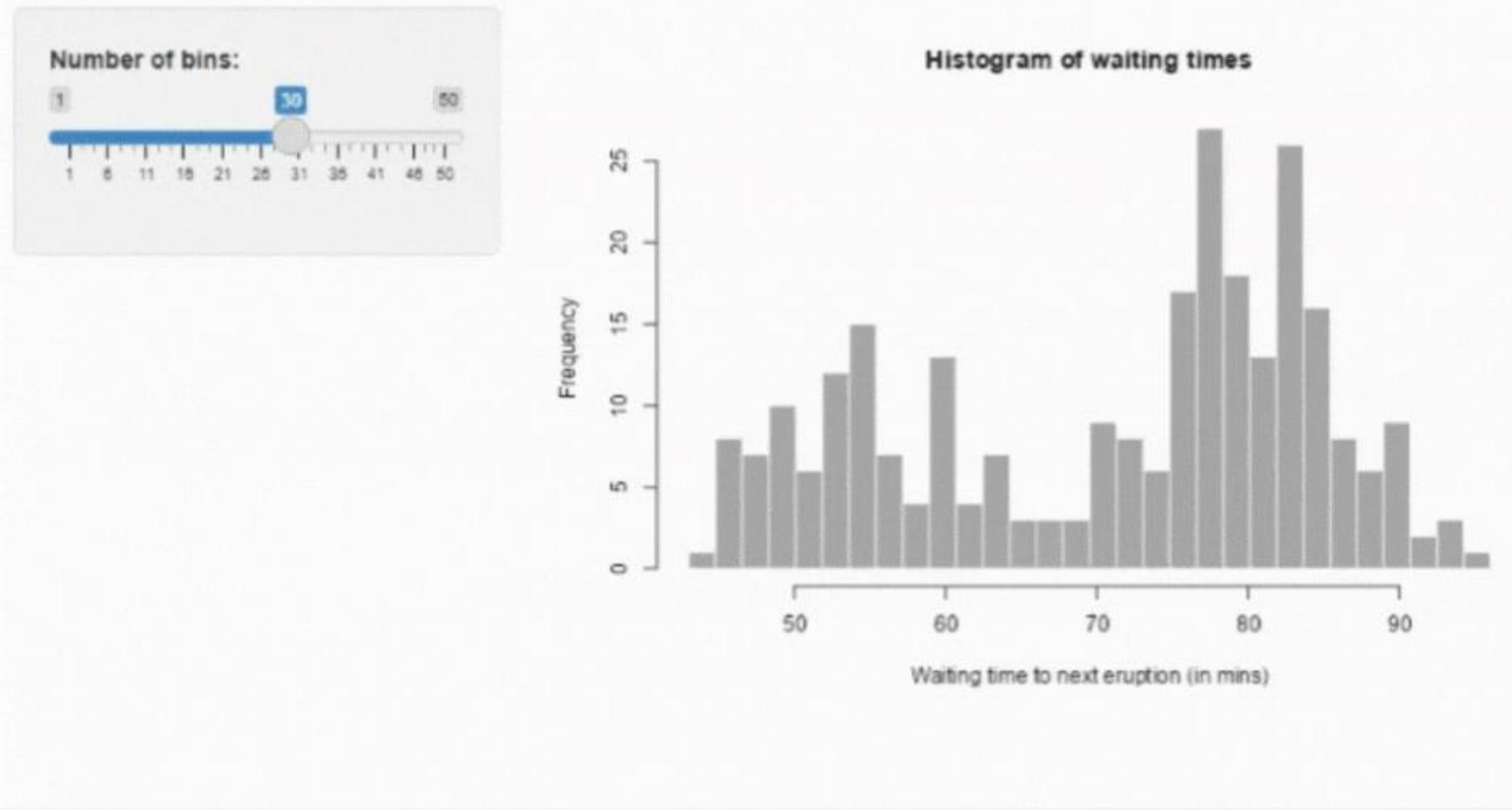
- movies.Rdata
- app.R

2. Open R Studio and create a new file as below, name it anything:



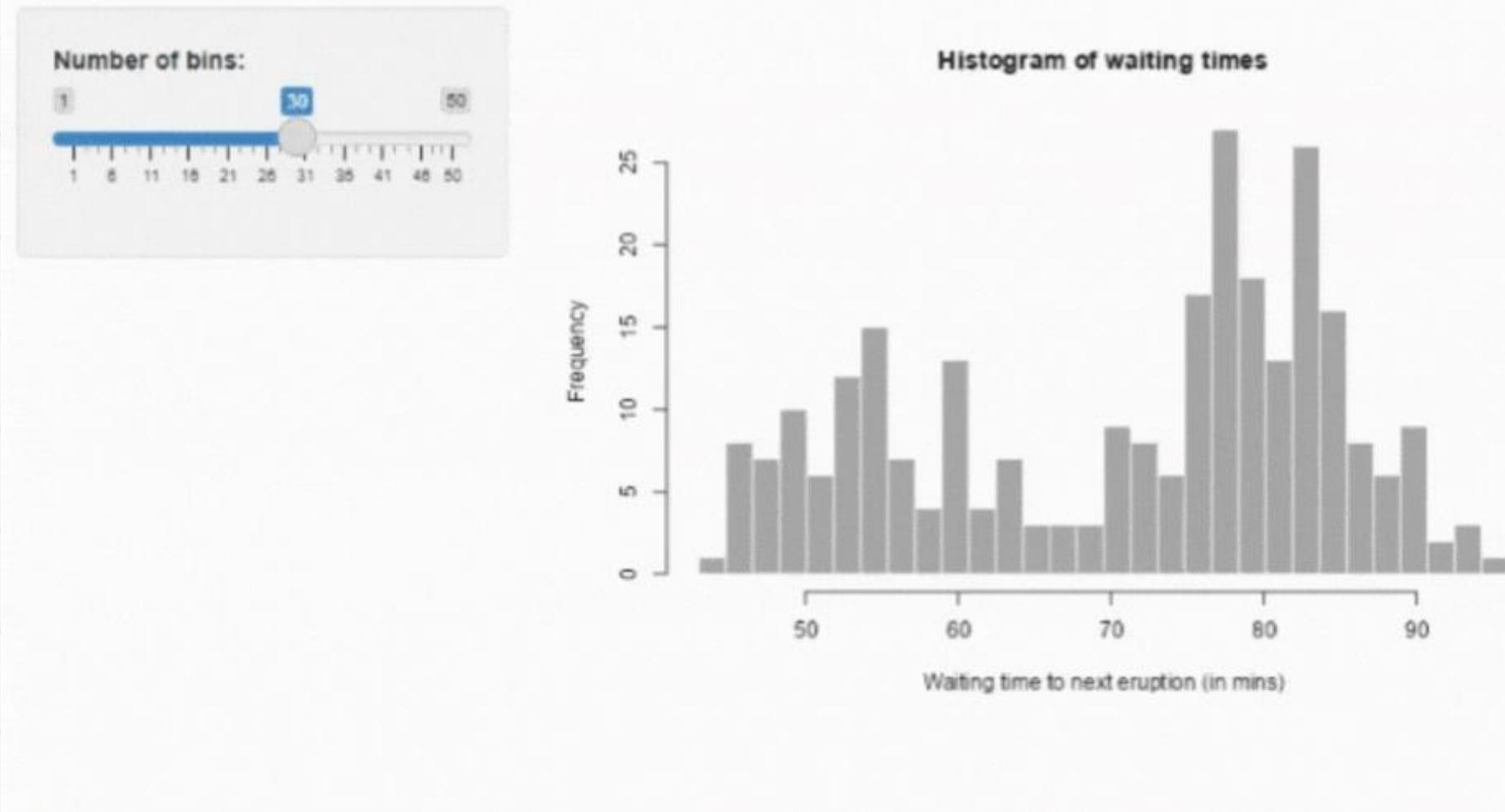
Exercise 1

Old Faithful Geyser Data



Exercise 1

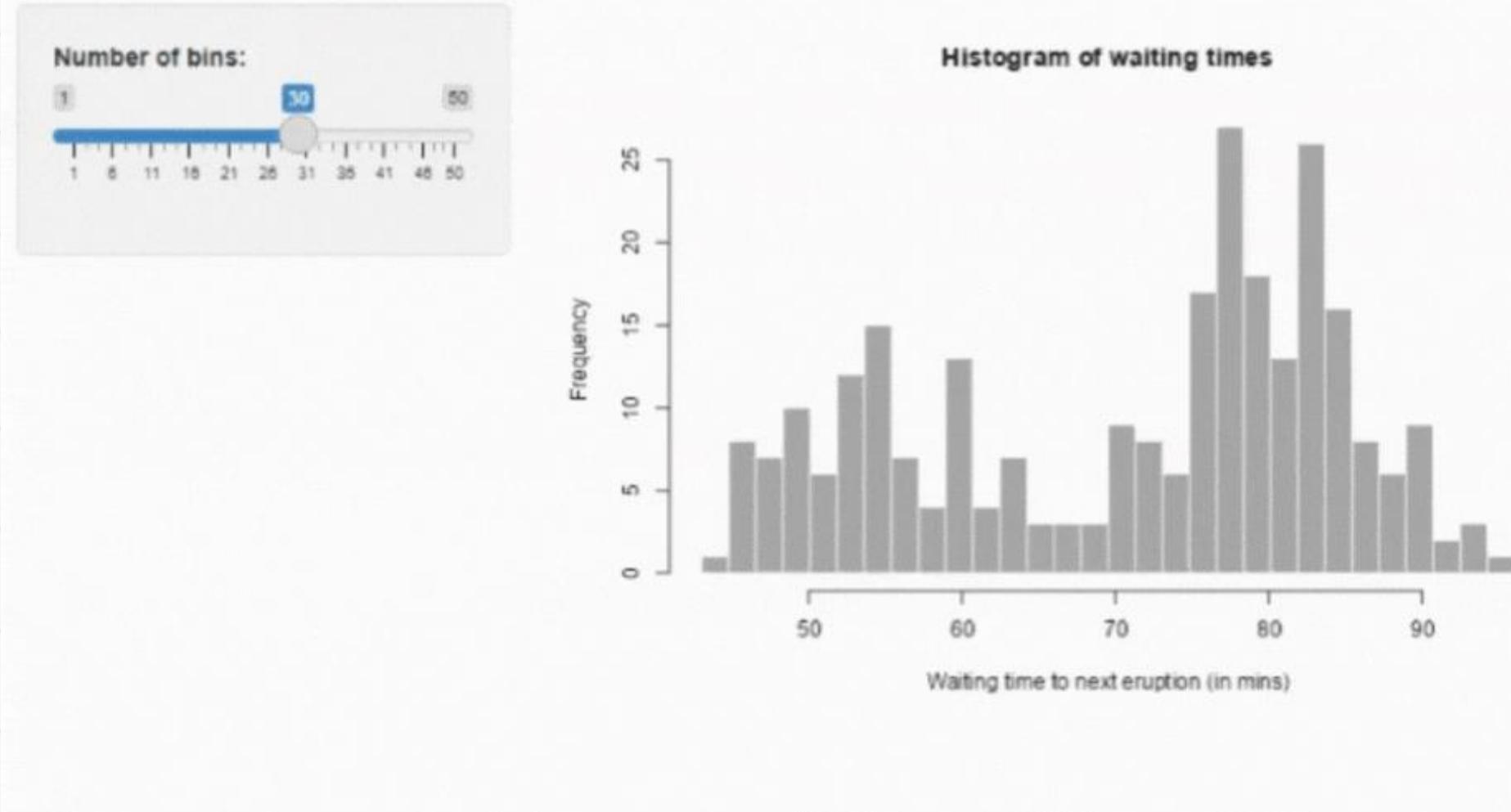
Old Faithful Geyser Data



1. Change the app title to “Hello World!”
2. Set the minimum value of the slider bar to 5
3. Change the histogram border color from "white" to "orange"
4. Change the sliderInput to a numericInput, with the same id and label and value = 30

Exercise 1

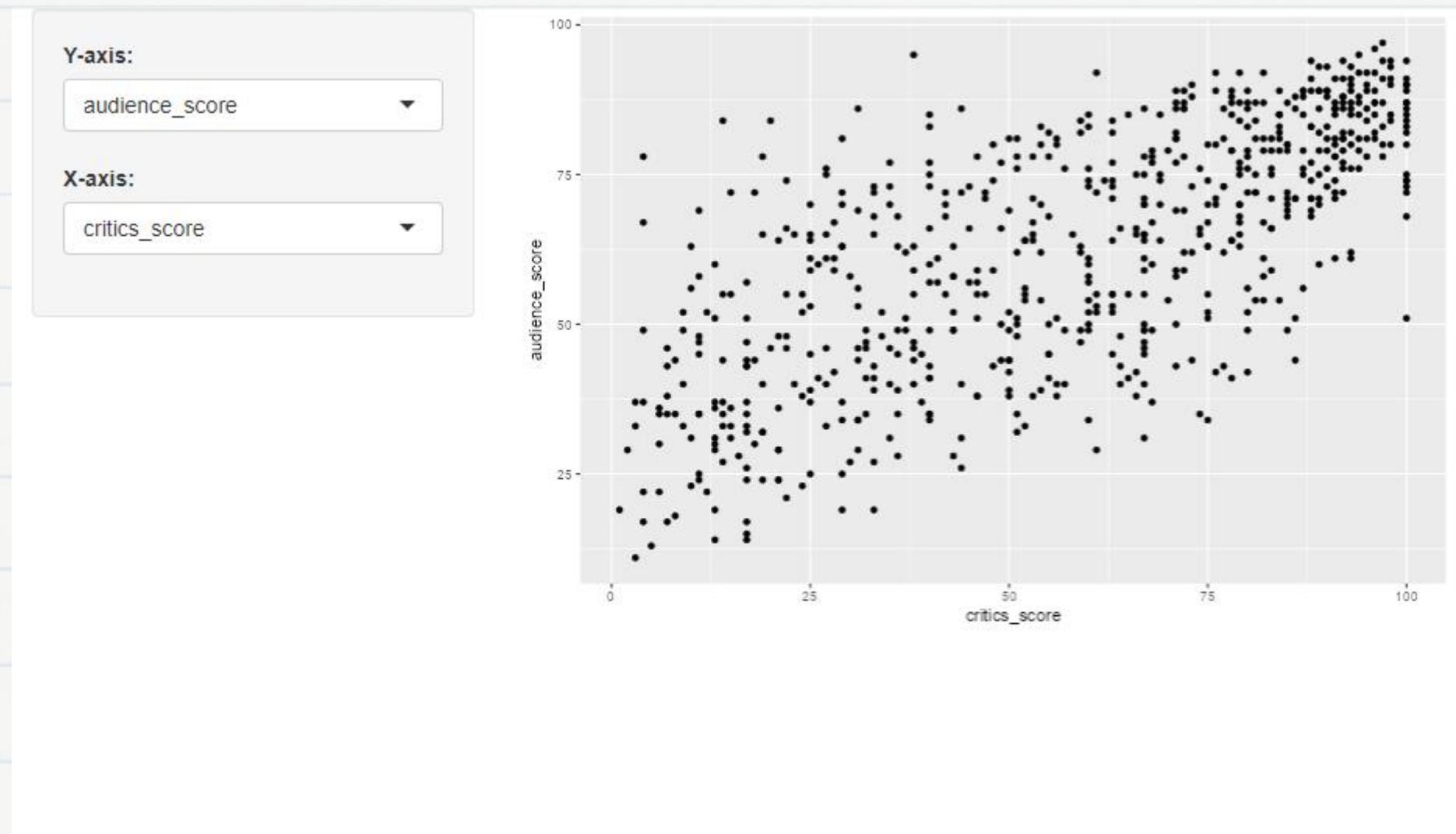
Old Faithful Geyser Data



1. Change the app title to “Hello World!”
2. Set the minimum value of the slider bar to 5
3. Change the histogram border color from "white" to "orange"
4. Change the sliderInput to a numericInput, with the same id and label and value = 30

```
numericInput(inputId = "bins",
             label = h3("Number of bins"),
             value = 30)
```

Exercise 2



- Make sure the `app.R` & `movies.RData` files are saved in the same folder.

Exercise 2

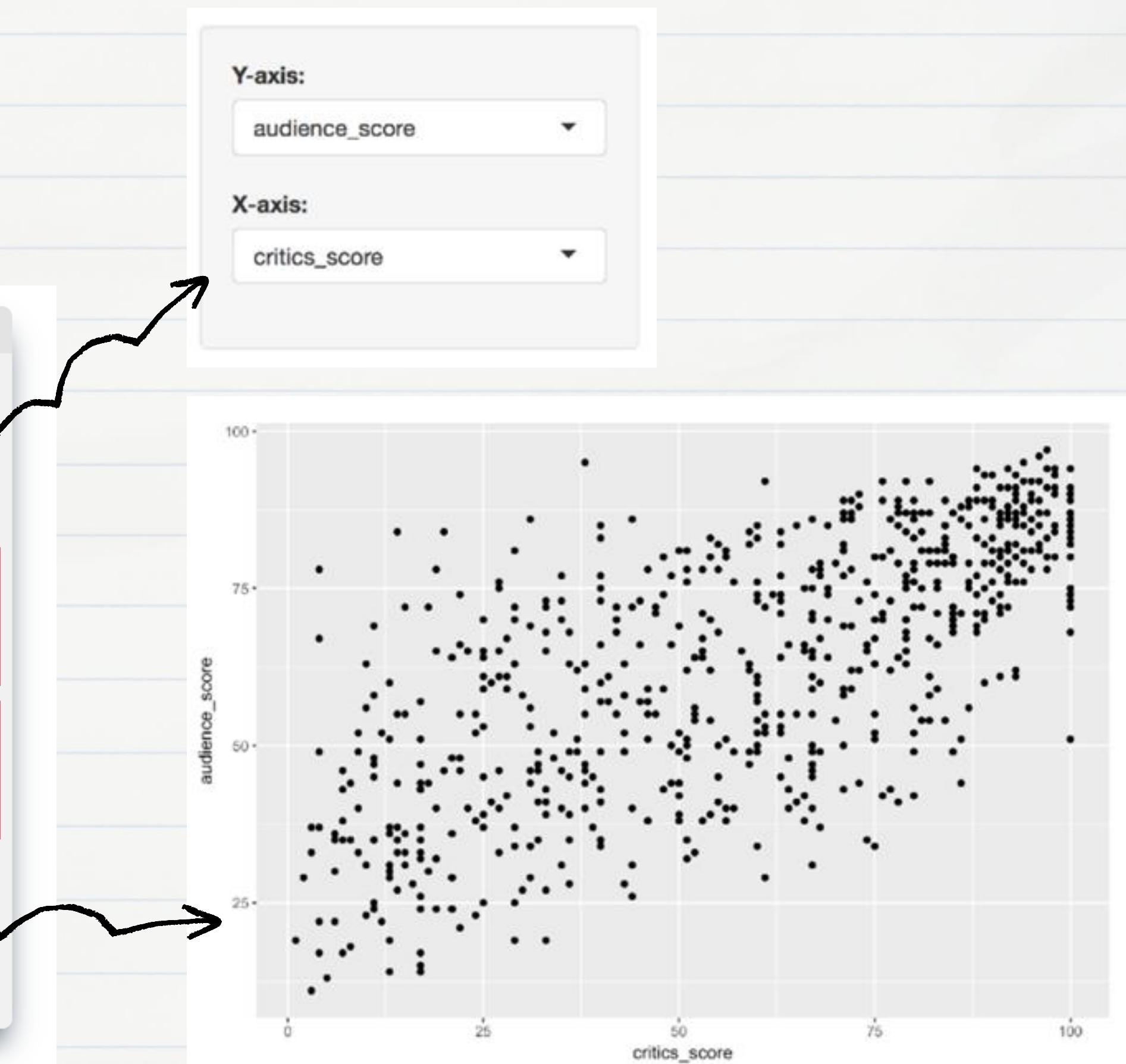


- Make sure the `app.R` & `movies.RData` files are saved in the same folder.

Exercise 2

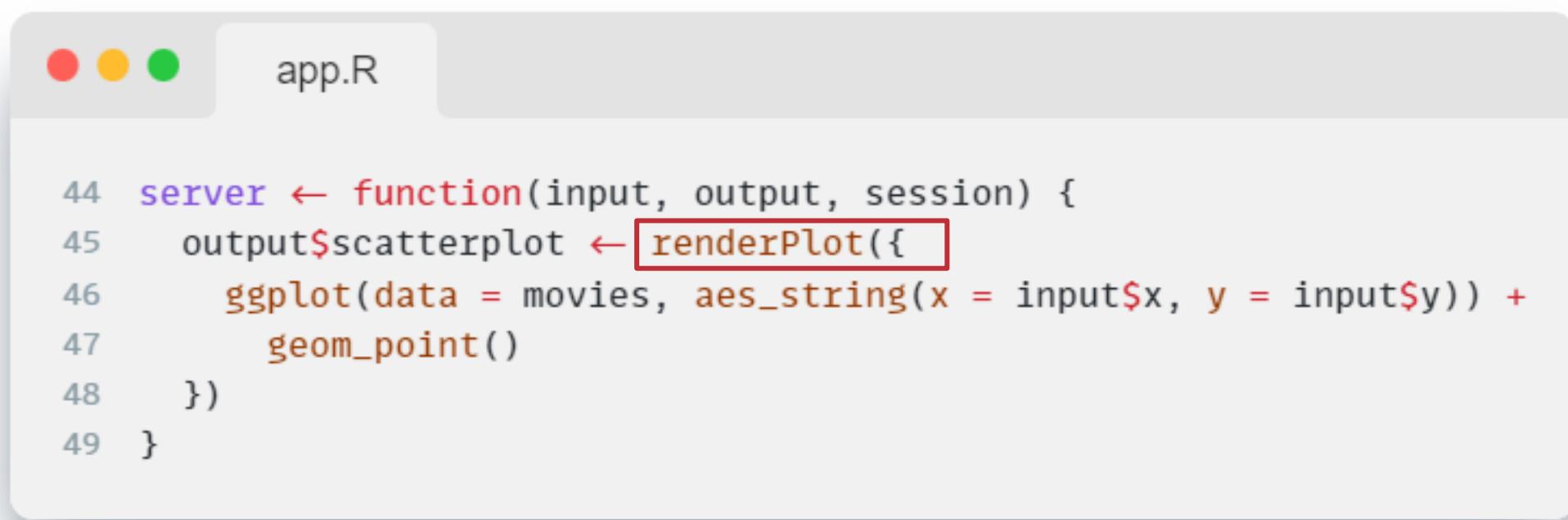
- Understand the UI

```
12 ui <- fluidPage(  
13  
14 sidebarLayout(  
15  
16 # Inputs: Select variables to plot  
17 sidebarPanel(  
18  
19 # Select variable for y-axis  
20 selectInput(  
21   inputId = "y",  
22   label = "Y-axis:",  
23   choices = c("imdb_rating", "imdb_num_votes", "critics_score", "audience_score", "runtime"),  
24   selected = "audience_score"  
25 ),  
26 # Select variable for x-axis  
27 selectInput(  
28   inputId = "x",  
29   label = "X-axis:",  
30   choices = c("imdb_rating", "imdb_num_votes", "critics_score", "audience_score", "runtime"),  
31   selected = "critics_score"  
32 )  
33 ),  
34  
35 # Output: Show scatterplot  
36 mainPanel(  
37   plotOutput(outputId = "scatterplot")  
38 )  
39 )  
40 )
```



Exercise 2

- Understand the server



app.R

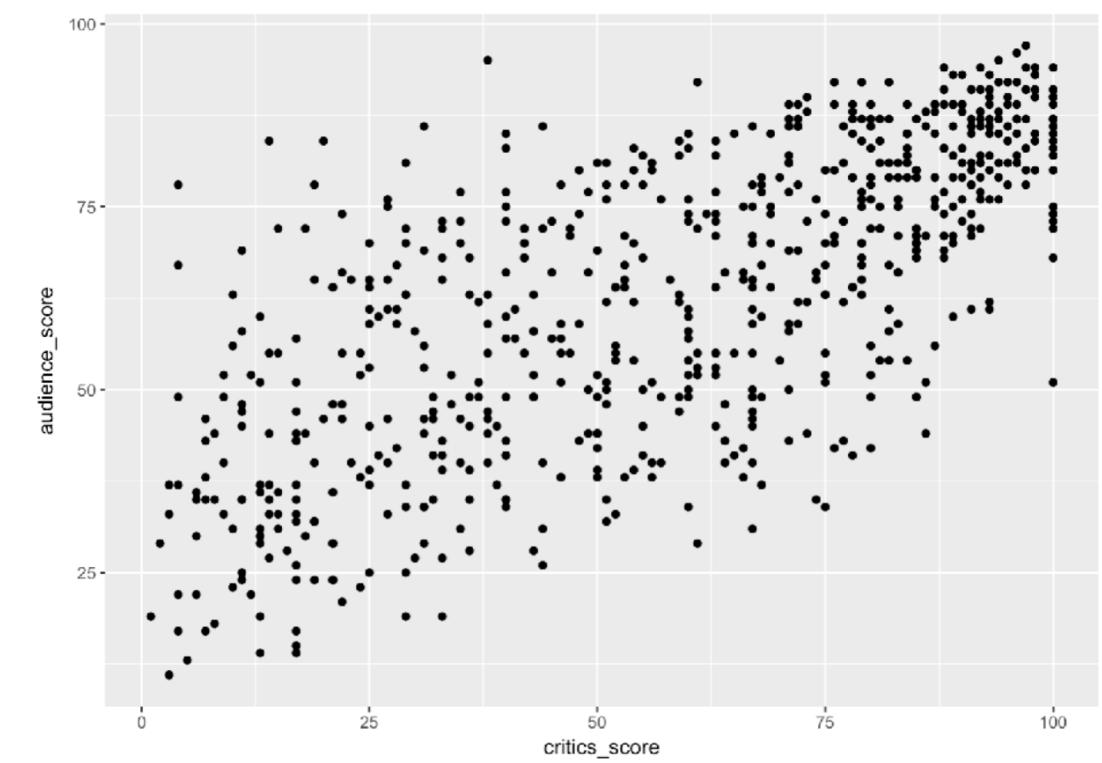
```
44 server <- function(input, output, session) {  
45   output$scatterplot <- renderPlot({  
46     ggplot(data = movies, aes_string(x = input$x, y = input$y)) +  
47       geom_point()  
48   })  
49 }
```

Y-axis:

audience_score

X-axis:

critics_score



Exercise 2 – part 1

In the UI

- Add a `selectInput()` to colour the points by a choice of the following variables:
`"title_type"`, `"genre"`, `"mpaa_rating"`, `"critics_rating"`, `"audience_rating"`
- Use `"z"` as the inputId

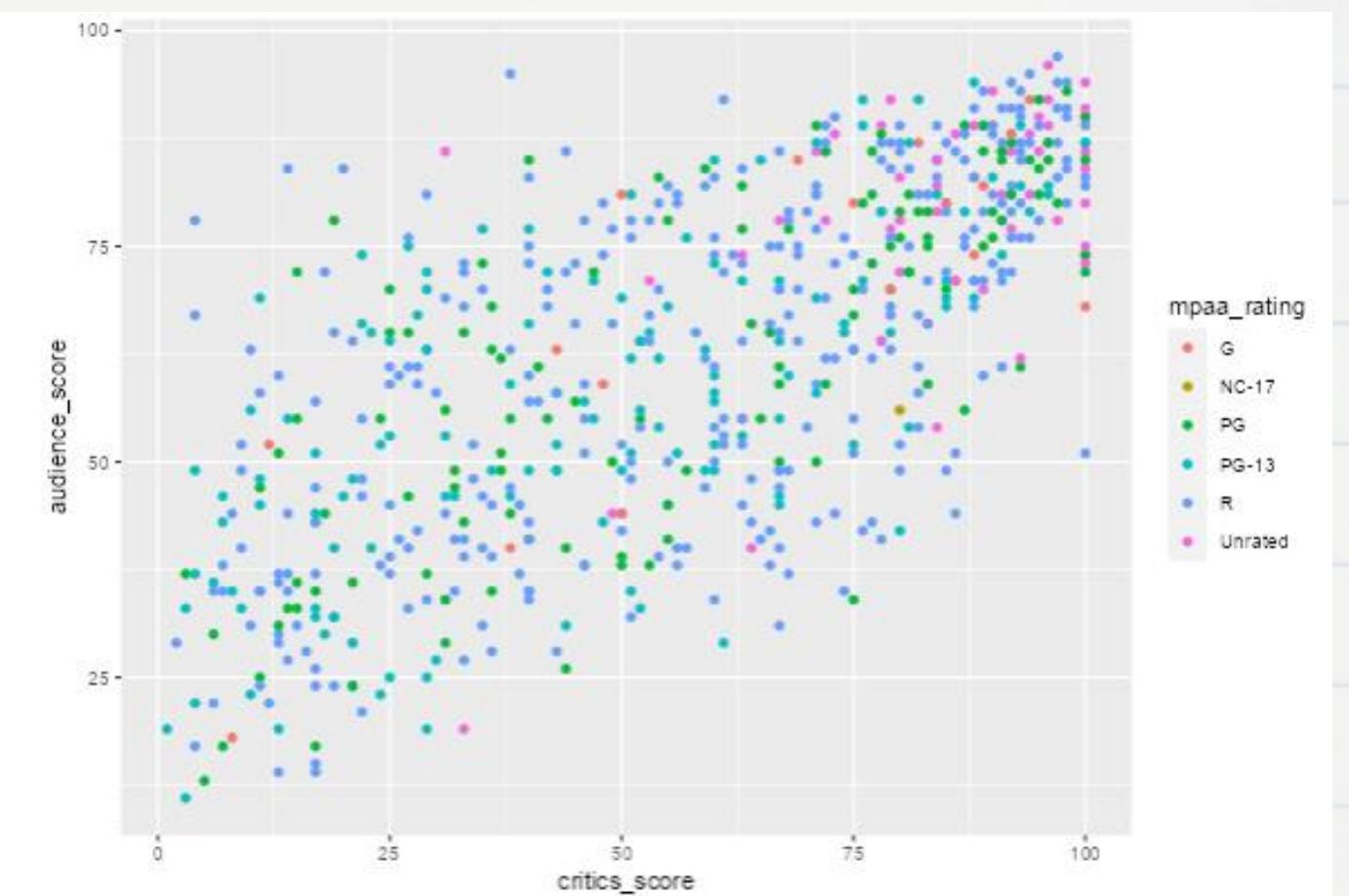
In the server

- Set the colour argument
in `ggplot()` as `color = input$z`

Y-axis: audience_score

X-axis: critics_score

Color by: mpaa_rating



Your app should look like this after

Exercise 2 – part 2

In the UI

- Add a `checkboxInput()`
- Add a `dataTableOutput()` below the plot output, where a table will appear when user clicks the check box

In the server

- Add a `renderDataTable` reactive expression that creates the table **if the checkbox is checked**
- The table should show the data from `movies.RData`

Your app should look like this after



The figure shows a Shiny application interface. At the top, there are dropdown menus for 'X-axis:' (set to 'Critics score') and 'Color by:' (set to 'MPAA rating'). Below these is a checkbox labeled 'Show data table' which is checked and highlighted with a red border. Underneath the checkbox is a data table with the following columns: title, title_type, genre, runtime, mpaa_rating, and studio. The table contains six rows of movie data:

title	title_type	genre	runtime	mpaa_rating	studio
Filly Brown	Feature Film	Drama	80	R	Indomi Media
The Dish	Feature Film	Drama	101	PG-13	Warne Picture
Waiting for Guffman	Feature Film	Comedy	84	R	Sony F Classic
The Age of Innocence	Feature Film	Drama	139	PG	Colum Picture
Malevolence	Feature Film	Horror	90	R	Ancho Enterta
Old Partner	Documentary	Documentary	78	Unrated	Shcalo Group
Lady Jane	Feature Film	Drama	142	PG-13	Param Home

Exercise 2 – part 3

In the UI

- Add an **input** widget that the user can interact with to **check boxes for selected title types**

In the server

- Add a **reactive expression** that subsets the data following these steps:
 - I. **group** the data by “title_type” and “genre”
 - II. **summarise** “imdb_rating” by mean, standard deviation and count, round the numbers
 - III. **filter** the data based on the selected title types

Select title type:

- Documentary
- Feature Film
- TV Movie

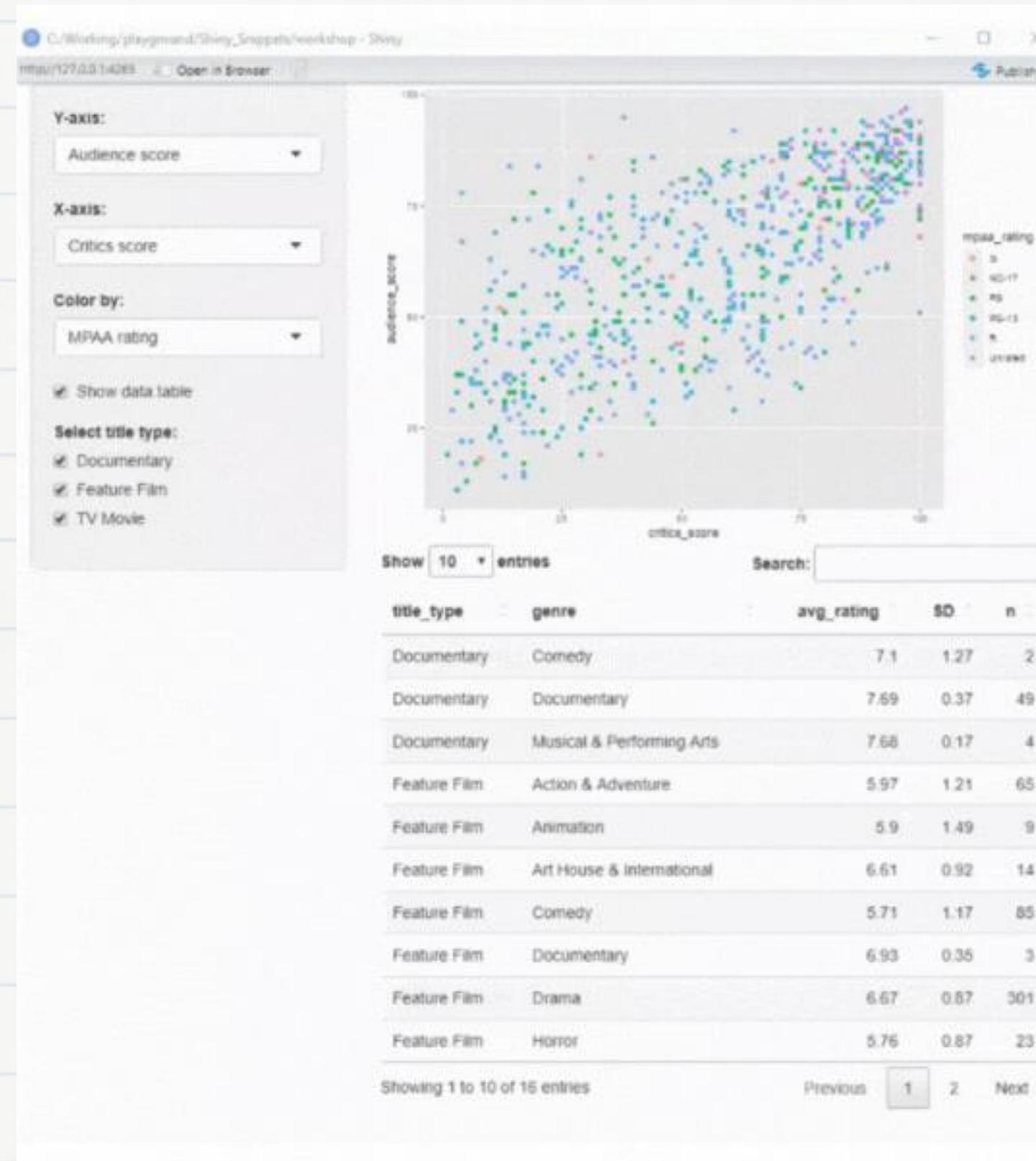
title_type	genre	avg_rating	SD	n
Documentary	Comedy	7.1	1.27	2
Documentary	Documentary	7.69	0.37	49
Documentary	Musical & Performing Arts	7.68	0.17	4
Feature Film	Action & Adventure	5.97	1.21	65
Feature Film	Animation	5.9	1.49	9
Feature Film	Art House & International	6.61	0.92	14
Feature Film	Comedy	5.71	1.17	85
Feature Film	Documentary	6.93	0.35	3
Feature Film	Drama	6.67	0.87	301
Feature Film	Horror	5.76	0.87	23

Showing 1 to 10 of 16 entries

Previous 1 2 Next

Your app should look like this after

Exercise 2 – recap



Improvements

- ✓ app title
- ✓ show/hide “select title type”
- ✓ interactive graph
- ✓ ...

Tricks and tips /1



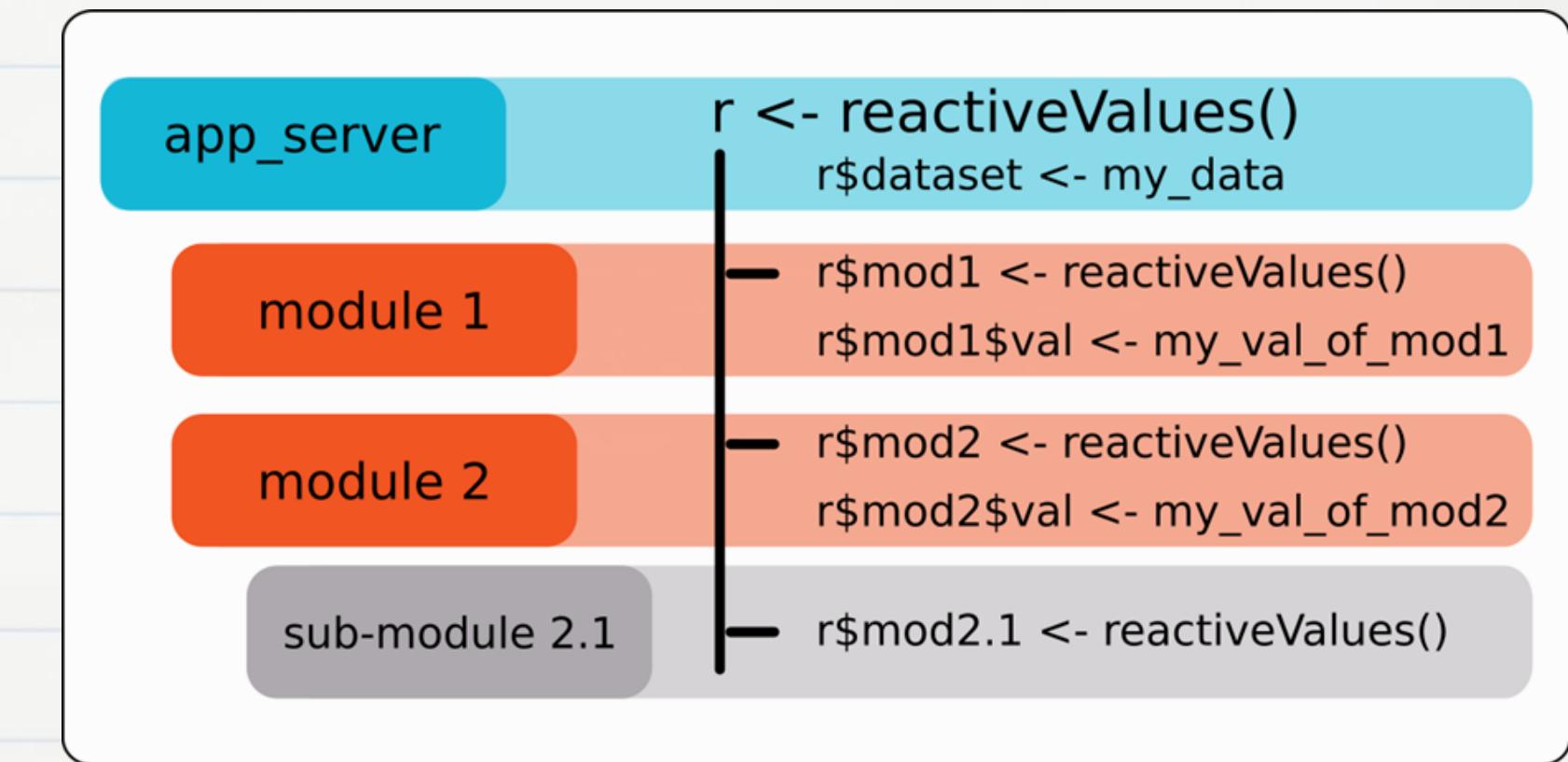
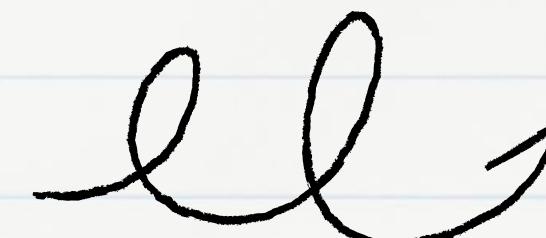
“Copy-and-paste is
a powerful tool,
but you should
avoid doing it
more than twice.”

— Hadley Wickham, Mine Çetinkaya-Rundel, and/or Garrett Grolemund,
R for Data Science (2e)

Tricks and tips /1

“Copy-and-paste is a powerful tool, but you should avoid doing it more than twice.”

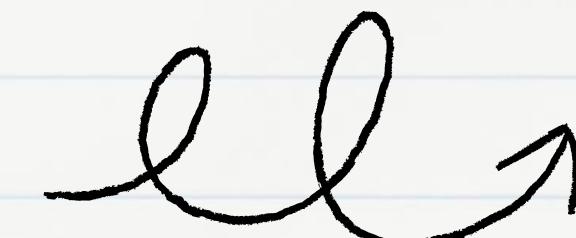
— Hadley Wickham, Mine Çetinkaya-Rundel, and/or Garrett Grolemund,
R for Data Science (2e)



Tricks and tips /1

“Copy-and-paste is a powerful tool, but you should avoid doing it more than twice.”

— Hadley Wickham, Mine Çetinkaya-Rundel, and/or Garrett Grolemund, R for Data Science (2e)



app_server

`r <- reactiveValues()`

`r$dataset <- my_data`

module 1

`- r$mod1 <- reactiveValues()`

`r$mod1$val <- my_val_of_mod1`

module 2

`- r$mod2 <- reactiveValues()`

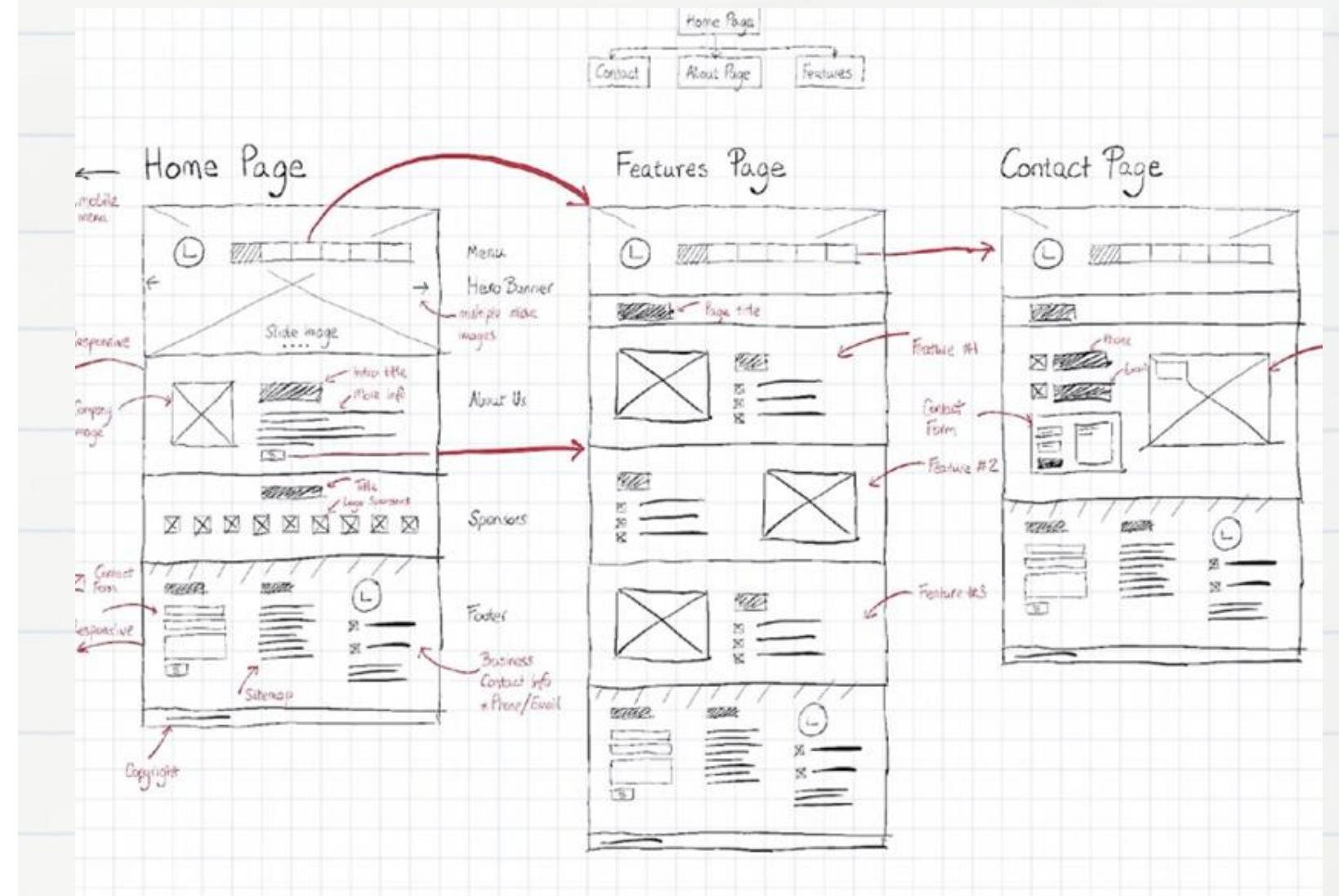
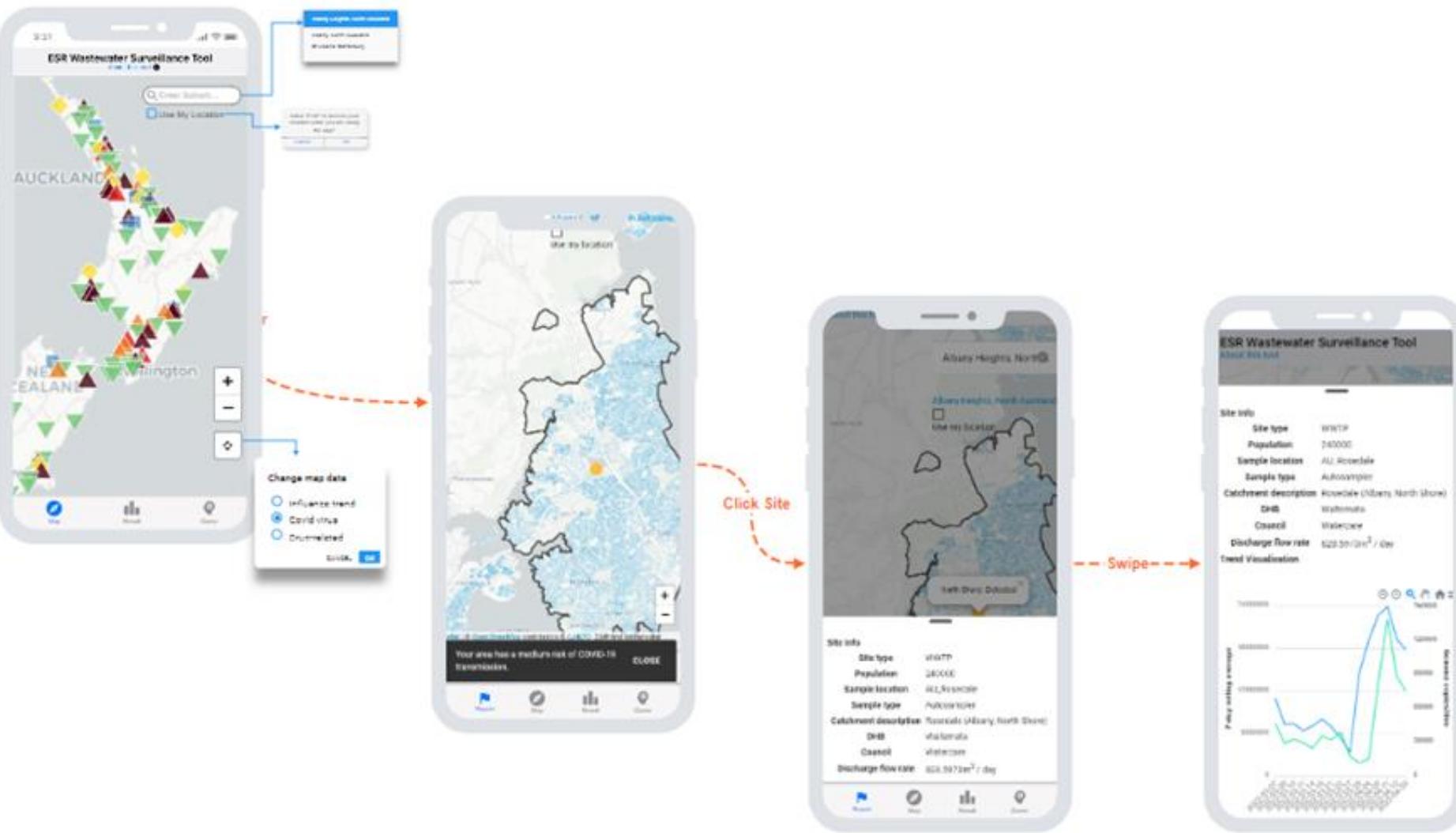
`r$mod2$val <- my_val_of_mod2`

sub-module 2.1

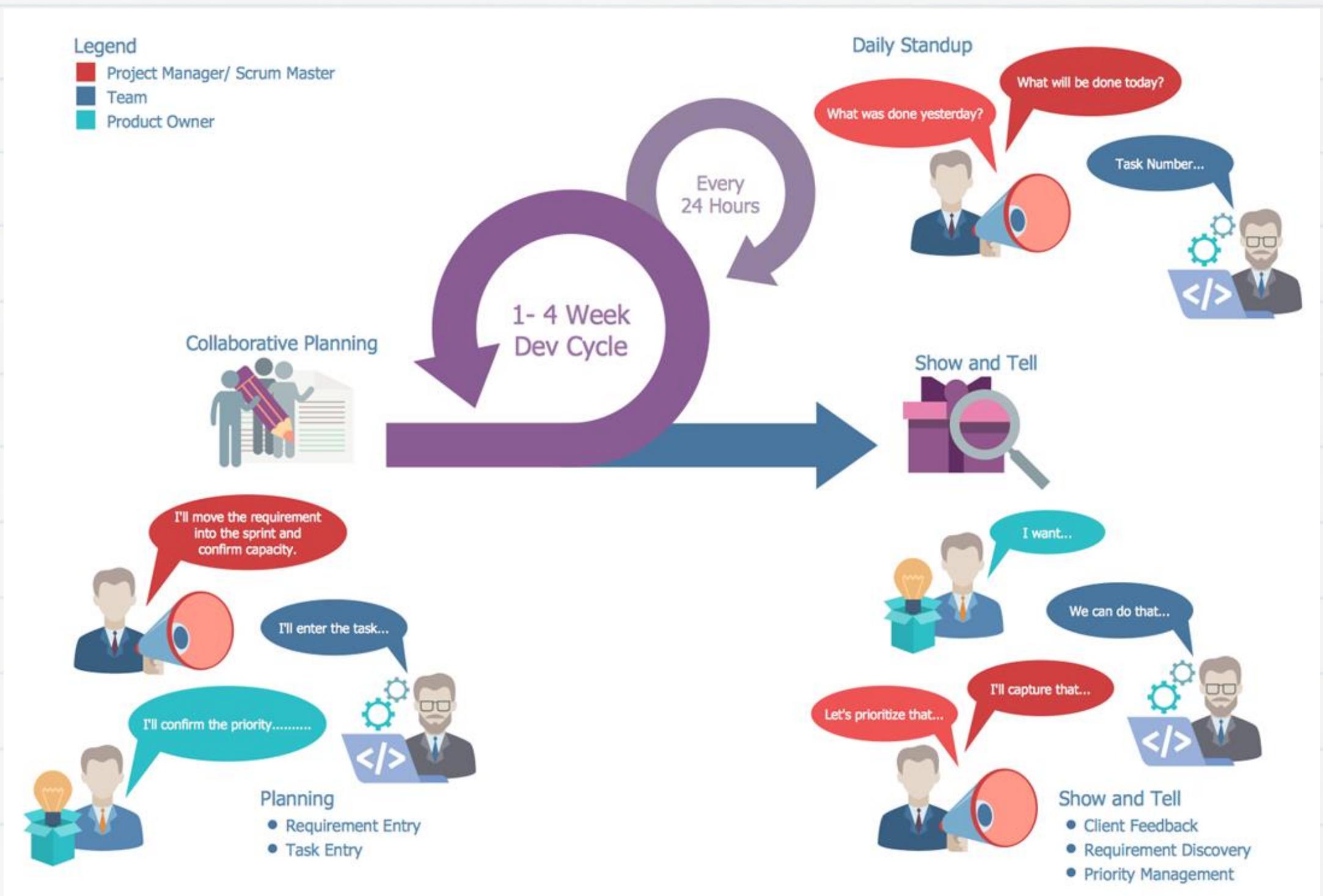
`- r$mod2.1 <- reactiveValues()`

Tricks and tips /2

Understand what you are building – wireframe design



Tricks and tips /3

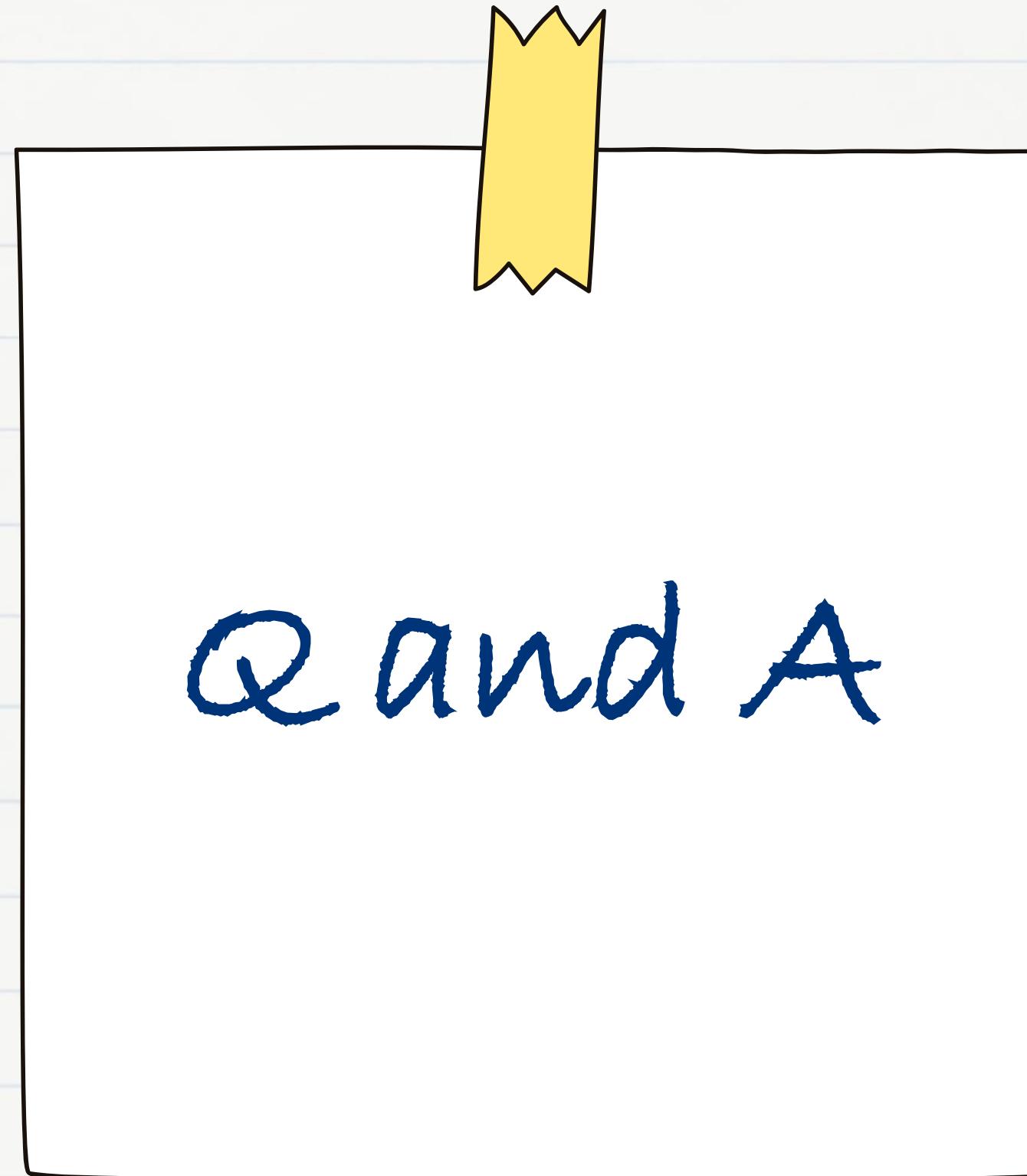




Resources



Type	Title	URL
Tutorial	Shiny official website	shiny.posit.co
News	Appsilon	appsilon.com
Community	R for Data Science	r4ds.io
Book	Mastering Shiny	mastering-shiny.org
Book	Engineering Production-Grade Shiny Apps	engineering-shiny.org
Book	JavaScript for R	book.javascript-for-r.com
Library	htmlwidgets	gallery.htmlwidgets.org
Library	R2D3	rstudio.github.io/r2d3



Lillian Lu
data + environment  diving + tourism 

