DATA MATTERS™

DATA SCIENCE SHORT COURSE SERIES

AUGUST 7 - 11, 2023

VIRTUAL VIA ZOOM

Welcome!
We'll be
starting soon!

UNC | ODUM INSTITUTE FOR RESEARCH IN SOCIAL SCIENCE

ncds | THE NATIONAL CONSORTIUM for DATA SCIENCE

renci

Advanced Visualization in R: R Shiny

Angela Zoss

Data Matters, Fall 2023

https://www.angelazoss.com/RShiny-1Day/

Slides and files

https://github.com/amzoss/RShiny-1Day

Schedule

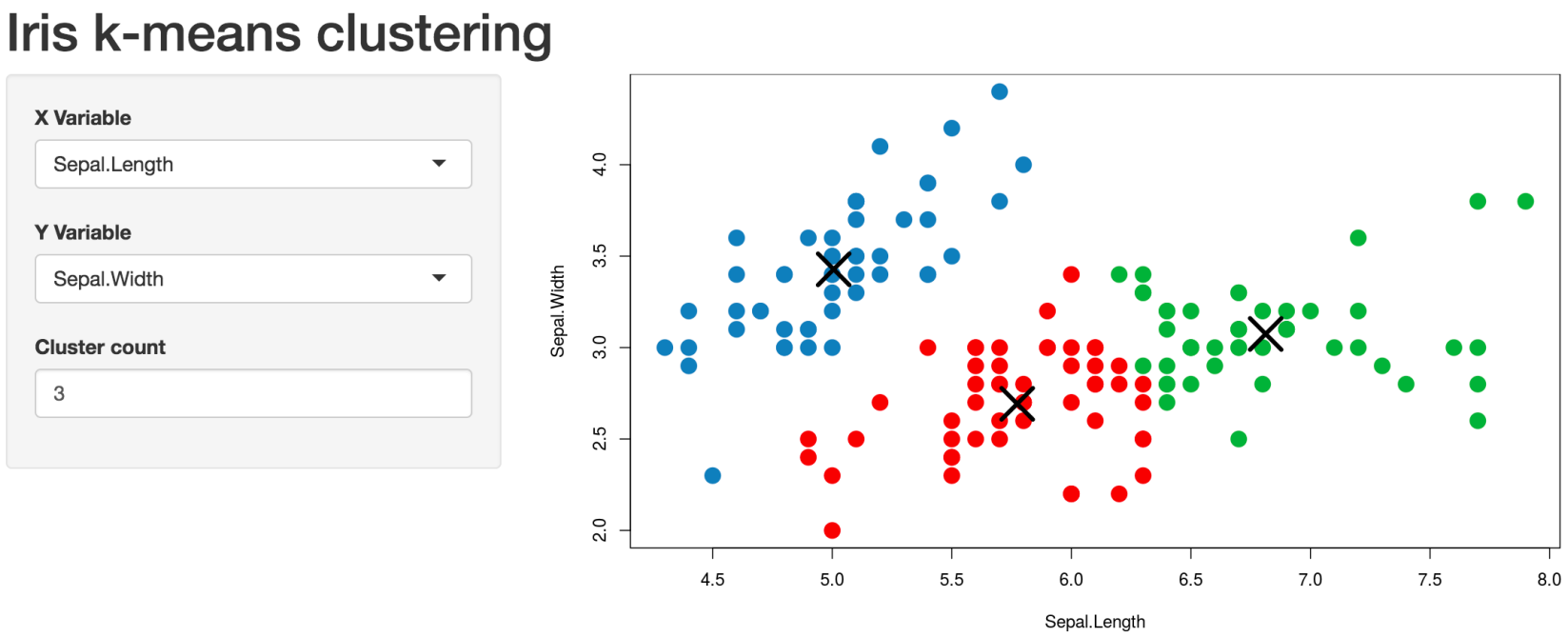
Session	Topics	Duration
Session 1	Introduction to data exploration, basic layout	9:30 a.m. – 10:35 a.m.
Morning break		10:35 a.m. – 10:50 a.m.
Session 2	Adding outputs, connecting basic inputs	10:50 a.m. – 11:55 a.m.
Lunch		11:55 a.m. – 1:10 p.m.
Session 3	Responding to user feedback, reactivity	1:10 p.m. – 2:15 p.m.
Afternoon break		2:15 p.m. – 2:30 p.m.
Session 4	Reducing duplication, using charts as inputs	2:30 p.m. – 3:35 p.m.
Q&A		3:35 p.m. – 3:40 p.m.

Other course logistics

- We all have different skill levels here. That's great!
- Questions and interruptions are welcome, especially if you are lost.
  I want everyone to be able to follow along.
- You may know the answer to someone else's question. If it's quick, feel free to make suggestions in the Zoom chat. Otherwise, I'll be happy to address it myself.
- You may have more advanced suggestions on top of what I'm teaching. Please try not to share these in chat. Too much chat can be a distraction, and I have a specific sequence I follow to keep the content approachable. You can share advanced things in Slack instead.

What is Shiny?

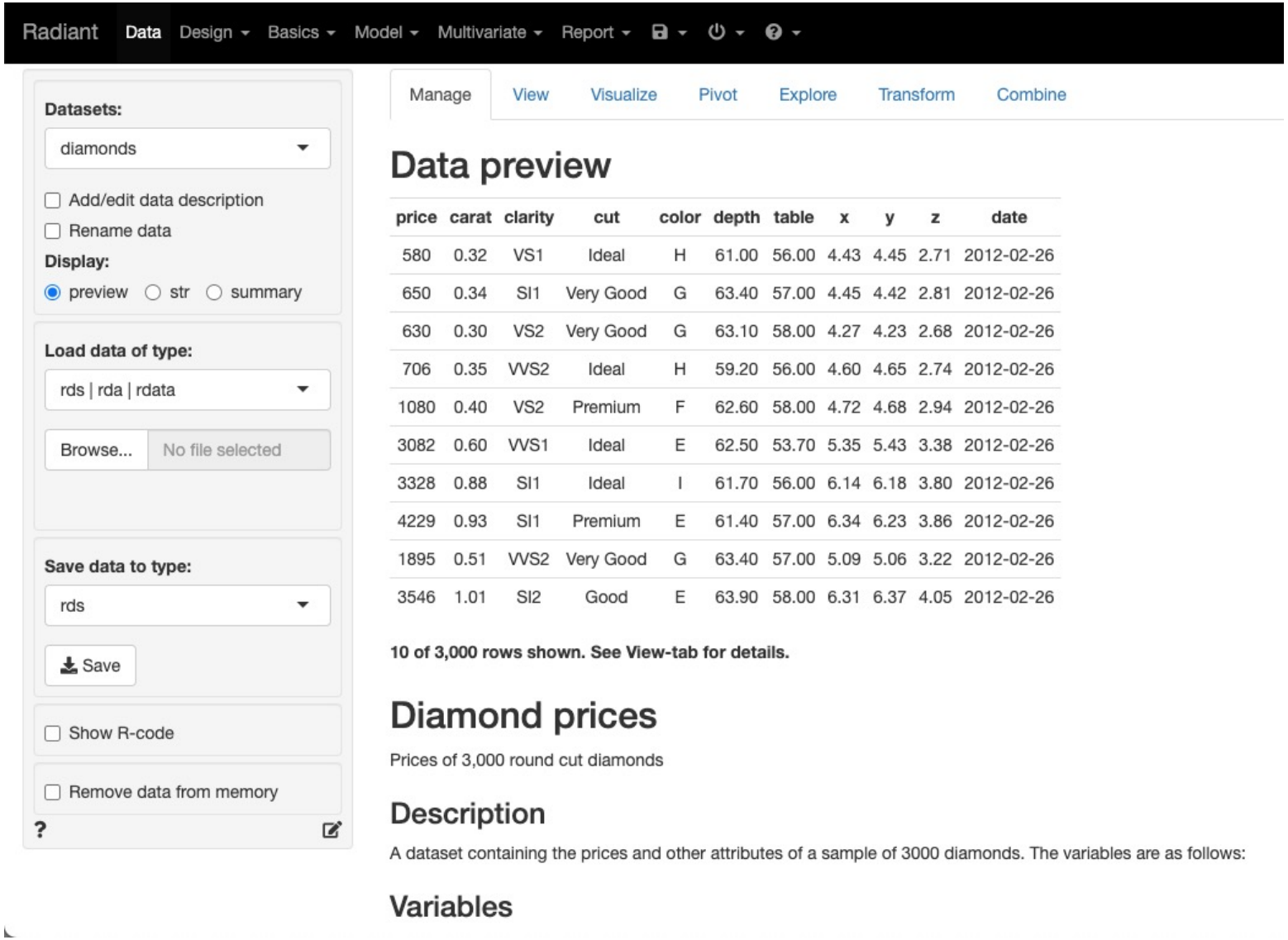
An interactive interface onto an R program



https://shiny.posit.co/

Why make a web app in R?

- Generalized workflows
- Custom subsetting
- Changing parameters
- Personalizing output



https://shiny.posit.co/r/gallery/education/radiant/

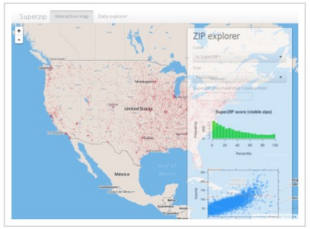
How does Shiny compare to other tools?

- **Static reports**: easy to share and very stable, but no exploration
- **Crosstalk package in R**: easy to publish and has some simple options for responding to user input, but quite limited
- **Tableau**: easy to build dashboards and respond to user input, but not reproducible

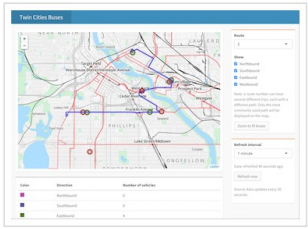
Shiny examples

Interactive visualizations

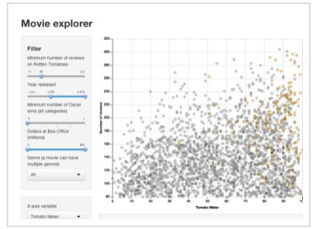
Shiny is designed for fully interactive visualization, using JavaScript libraries like d3, Leaflet, and Google Charts.



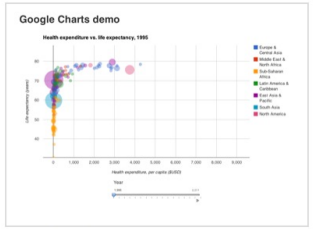
SuperZip example



Bus dashboard



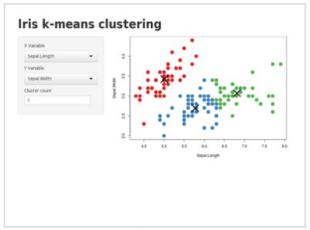
Movie explorer



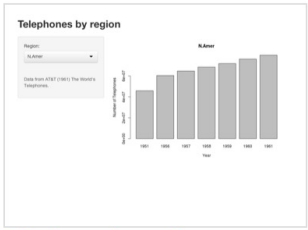
Google Charts

Start simple

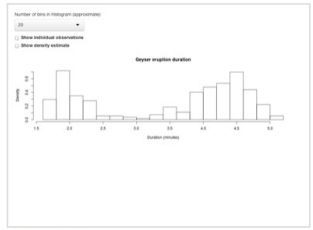
If you're new to Shiny, these simple but complete applications are designed for you to study.



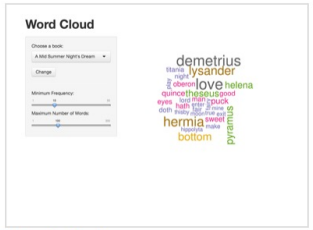
Kmeans example



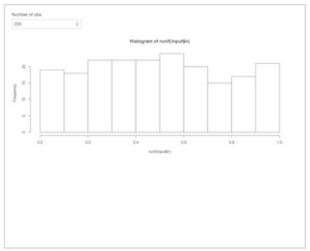
Telephones by region



Faithful



Word cloud



Single-file shiny app

https://shiny.posit.co/r/gallery/

Set up our environment

Get workshop files

URL: https://github.com/amzoss/RShiny-1Day

On GitHub:

- Click green “Code” button and select “Download ZIP”
- Unzip files on your computer
  - Windows: Double-click, then look for “Extract Files” at the top
  - Mac: Double-click
- Note: have noticed some issues when using OneDrive to store files

In RStudio:

- Project → New project…
- Existing directory
- Select unzipped folder
- Create Project

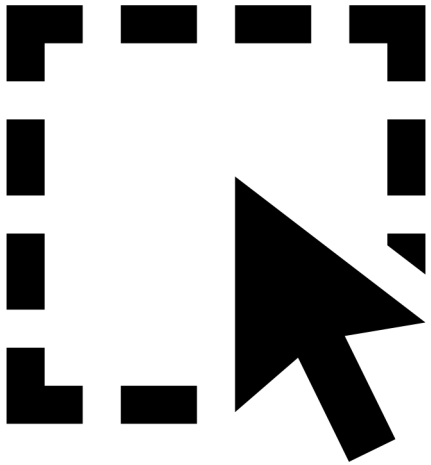
Shiny basics

Planning for Interactivity

Output

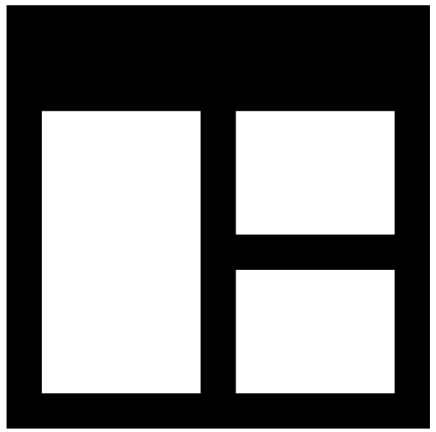
Picking the right visual elements

Input



Giving users the right controls

Layout



Arranging everything in the right place

How do you build a Shiny app?

- **Layout:** Setup the layout
- **Inputs and Outputs:** Insert visual components or placeholders
- **Server:** Write the R code to fill in the placeholders based on user feedback

Shiny code is split into two main parts

User Interface (UI)

the website people will see and interact with

Server

special R code that generates and updates content for the website

Running the app on your computer

- Develop the app in RStudio
- Click Run App
- View in RStudio or in browser

Sharing an app

- Shiny Apps (publish a small number of apps for free, pay for more) http://www.shinyapps.io/
- Shiny Server (host your own server for free) https://posit.co/download/shiny-server/
- Posit Connect (enterprise solution, high fees) https://posit.co/products/open-source/shinyserver/

Fixing Errors

Debugging code

- Start simple
- If you see an error:
  - read error message for hints
  - check for problems with spelling/punctuation marks
- Get code to run without errors
- Check result to see if it makes sense

- Add a small change
- Get code to run without errors
- Check result to see if it makes sense
- etc.

RStudio built-in help documentation

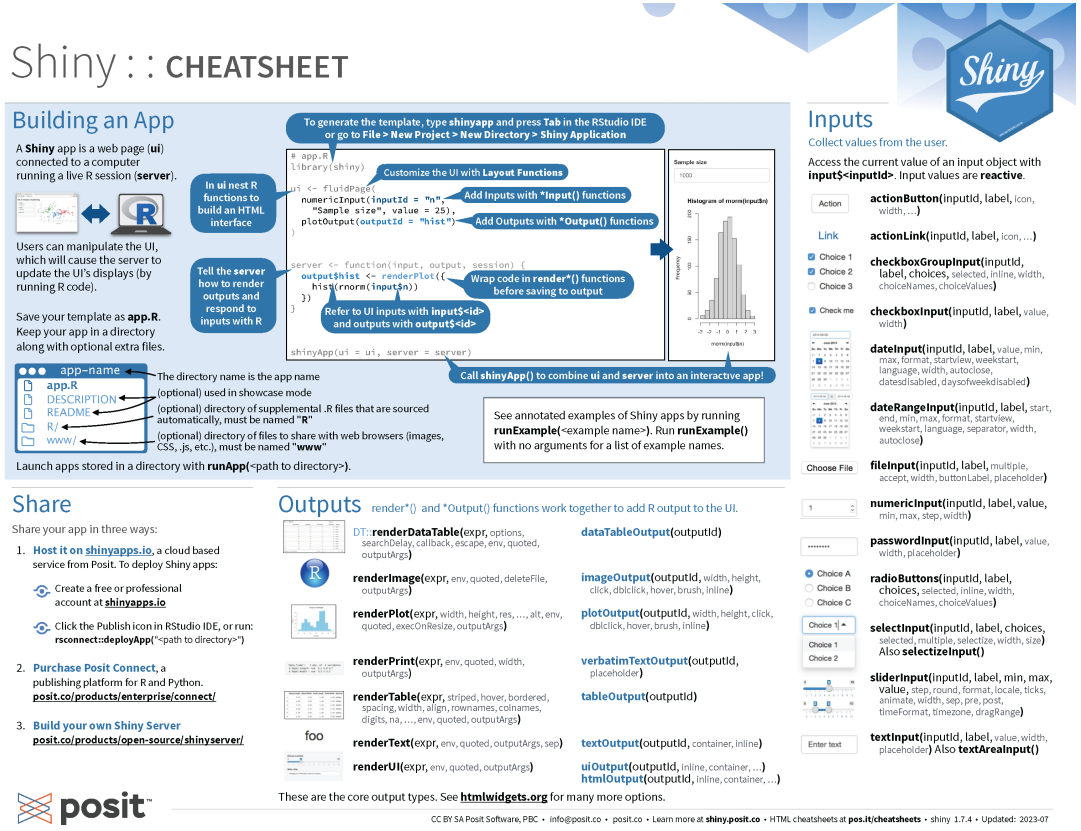
- In console, type
  `?<function or package name>`
- In help tab (not help menu), type into main search box
- In package tab, click on package name
- In help menu, use the Cheat Sheets submenu to download cheat sheet PDFs

Shiny Cheat Sheet

Help →

Cheatsheets →

Web Applications with shiny



Basic Layout

Mechanics of Shiny Layouts

- Build a page out of containers, which are all R functions
- Insert visual components into those functions as a list separated by commas

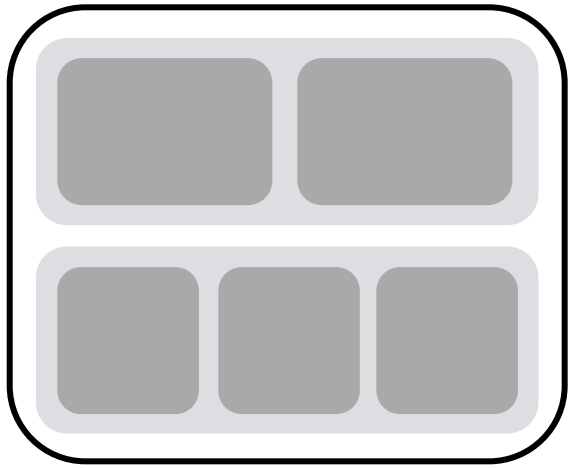
Examples:

```
layoutContainer(visualComponent(…), visualComponent2(…))
layoutContainer(
  visualComponent(…),
  layoutContainer(visualComponent2(…), visualComponent3(…))
)
```

Start with a Page

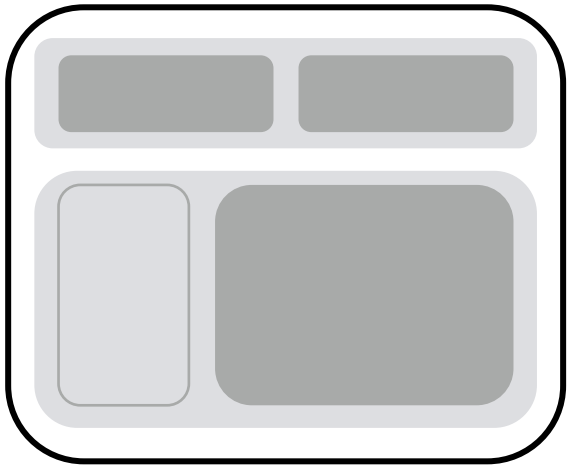
Fixed Page

- Most limited
- Can only arrange elements in rows, which then get split into columns



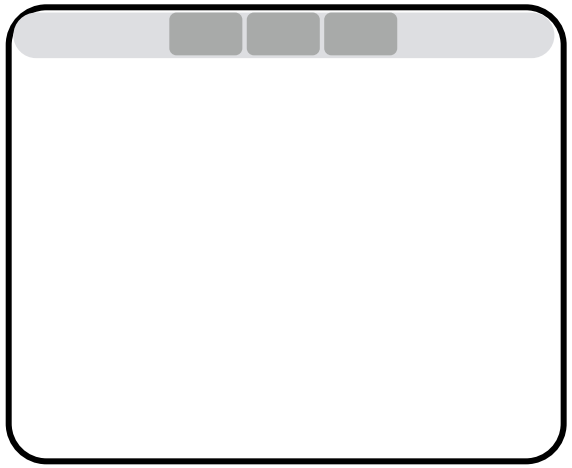
Fluid Page

- Can arrange elements in rows, which then get split into columns
- Can also include other layouts and panels



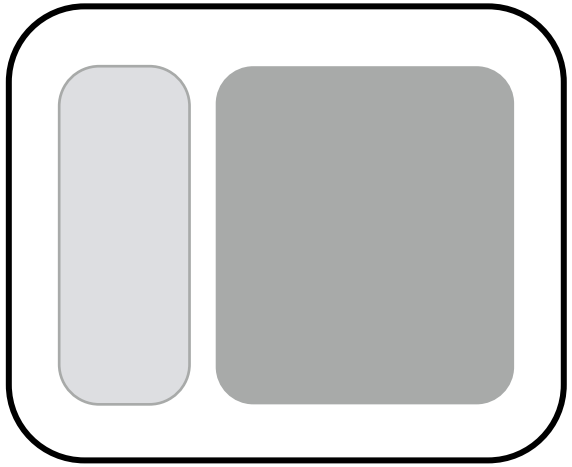
Navbar Page

- Creates a navigation bar at the top
- Can specify a title
- Can add multiple pages, each of which is fluid

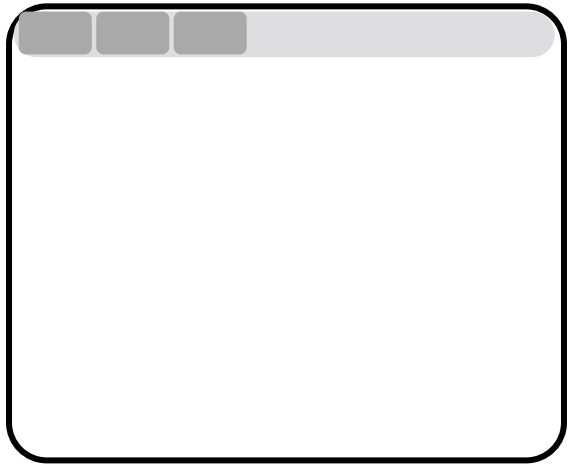


Common Layout Options

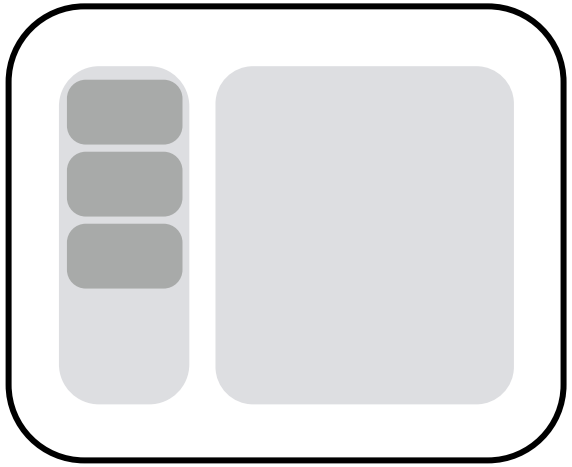
Sidebar Layout



Tabset Panel



Navlist Panel

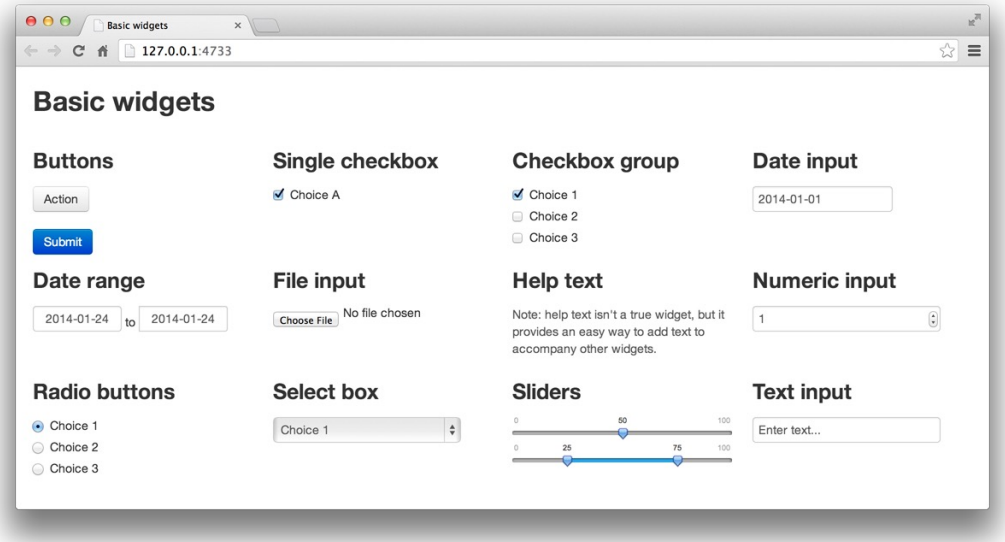


Additional Panels and Containers

- **Fixed or Fluid Row and Column**
  For row-based pages, can organize content into rows, then split into columns. Column widths go from 1 to 12.
- **Title Panel**
  Good way to provide a title to a page or tabset panel
- **Well Panel**
  Basically a colored box you can put content in
- **Conditional Panel**
  You can set this up so it is hidden until a condition is met

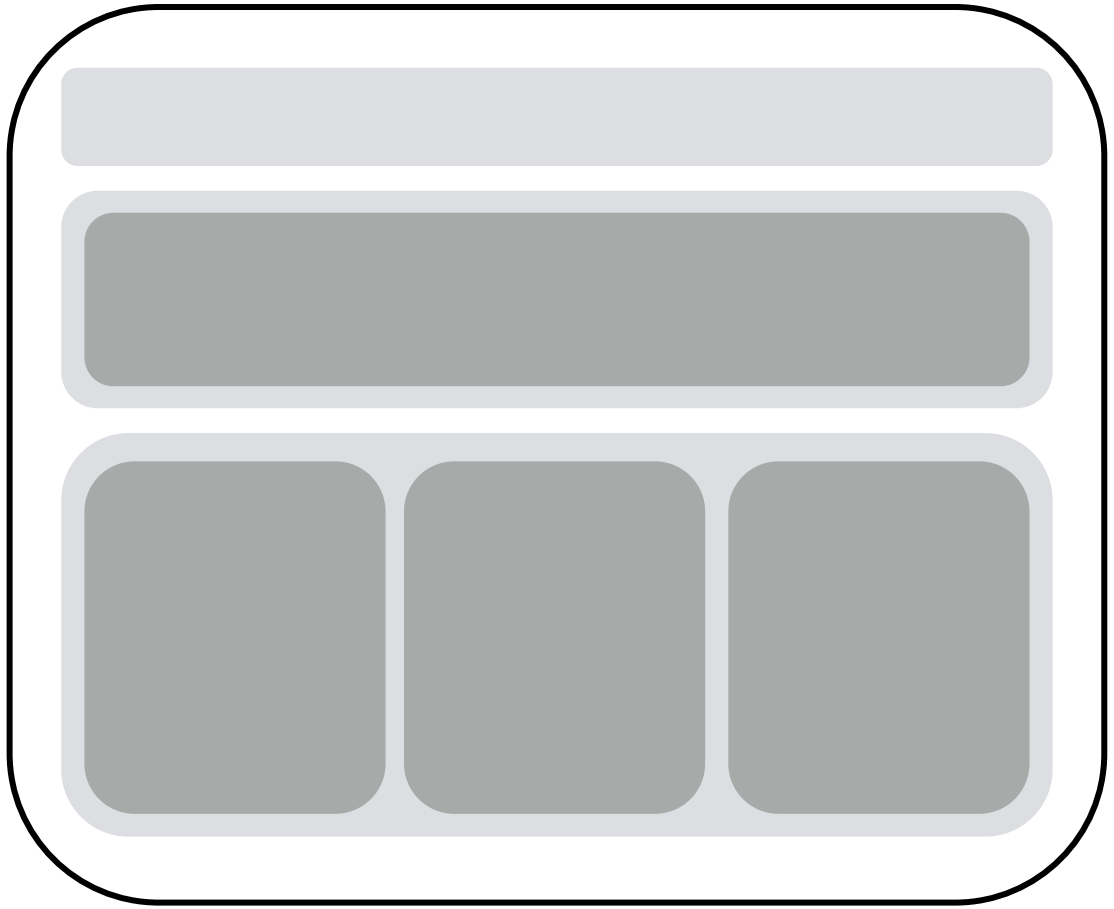
What goes in the containers?

- Simple **static content**
  (like headers, text, images)
- Some kind of **input widget**
  (a way to get user feedback)
- Some kind of **render object**
  (a display created by R code)



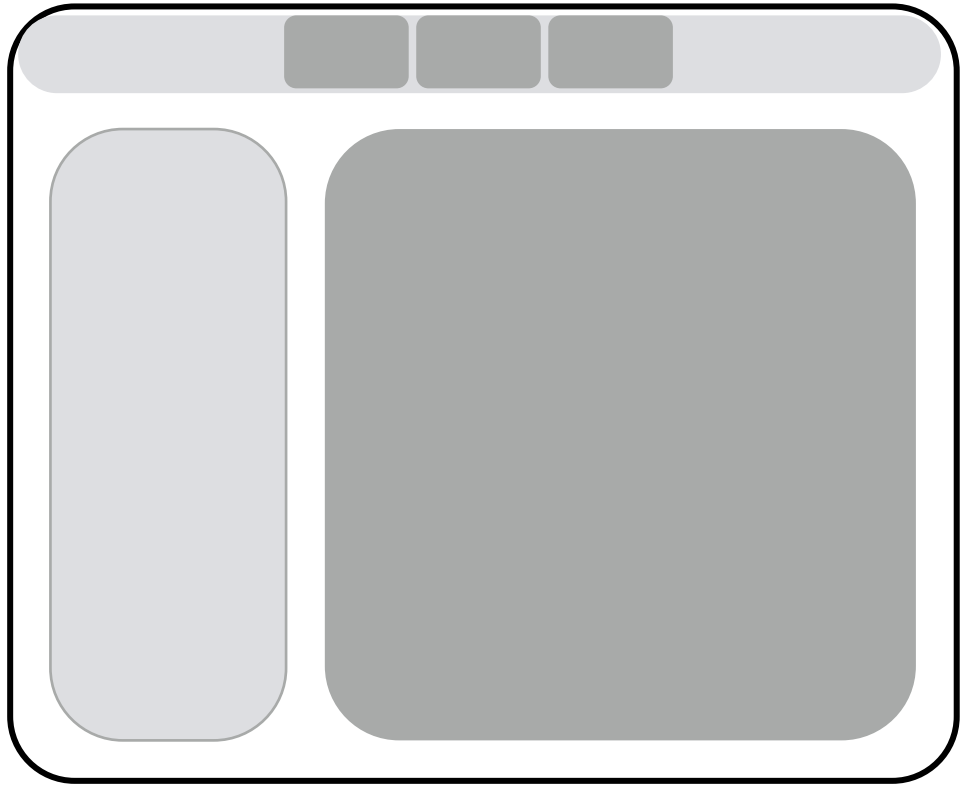
Exercise 1: Basic Layout

- Create a fluid page
- Add a title panel
- Add two fluid rows, one with a single column, one with three columns
- Add static content (text and images) to the columns



Exercise 2: Multi-Page Layout

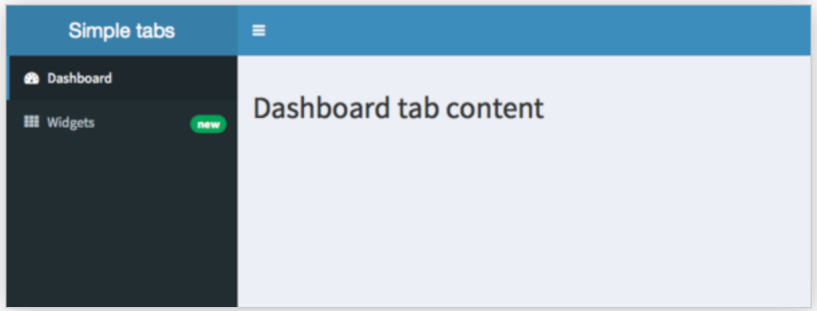
- Create a navbar page
- Give it a title
- Add four tab panels
- Use each to test a different layout:
  - Sidebar Layout
  - Tabset Panel
  - Navlist Panel
  - Your choice
- Add static content to different parts of the layout to see where it shows up



Different layout solution: Shiny Dashboard

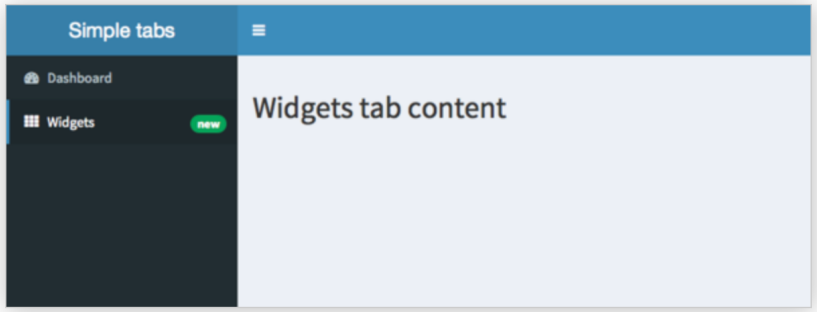
Sidebar menu items and tabs

Links in the sidebar can be used like `tabPanel`s from Shiny. That is, when you click on a link, it will display different content in the body of the dashboard. Here is an example of a simple tabPanel:



Tabs

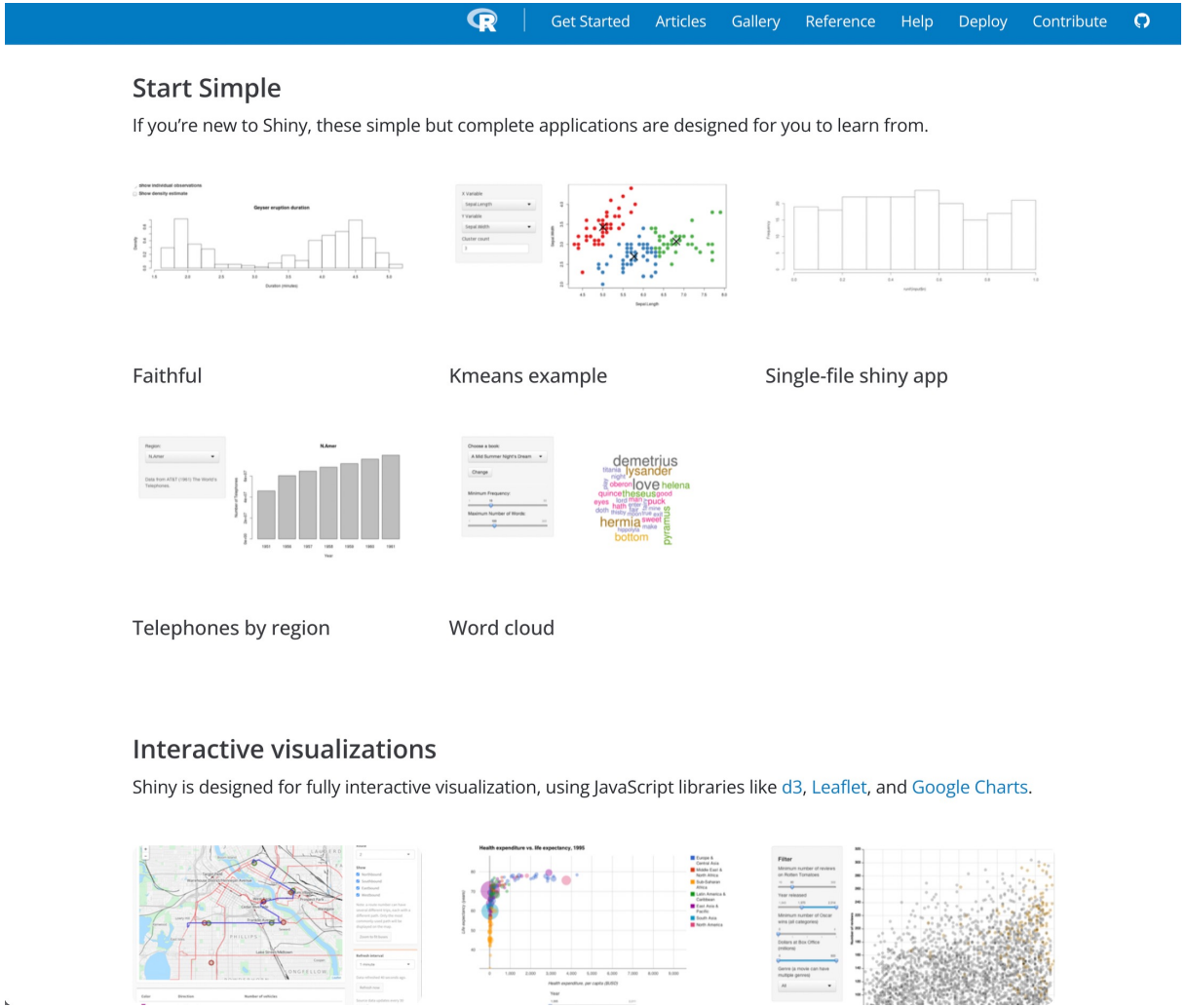
When the user clicks on one of the menu items, it switches content in the main body:



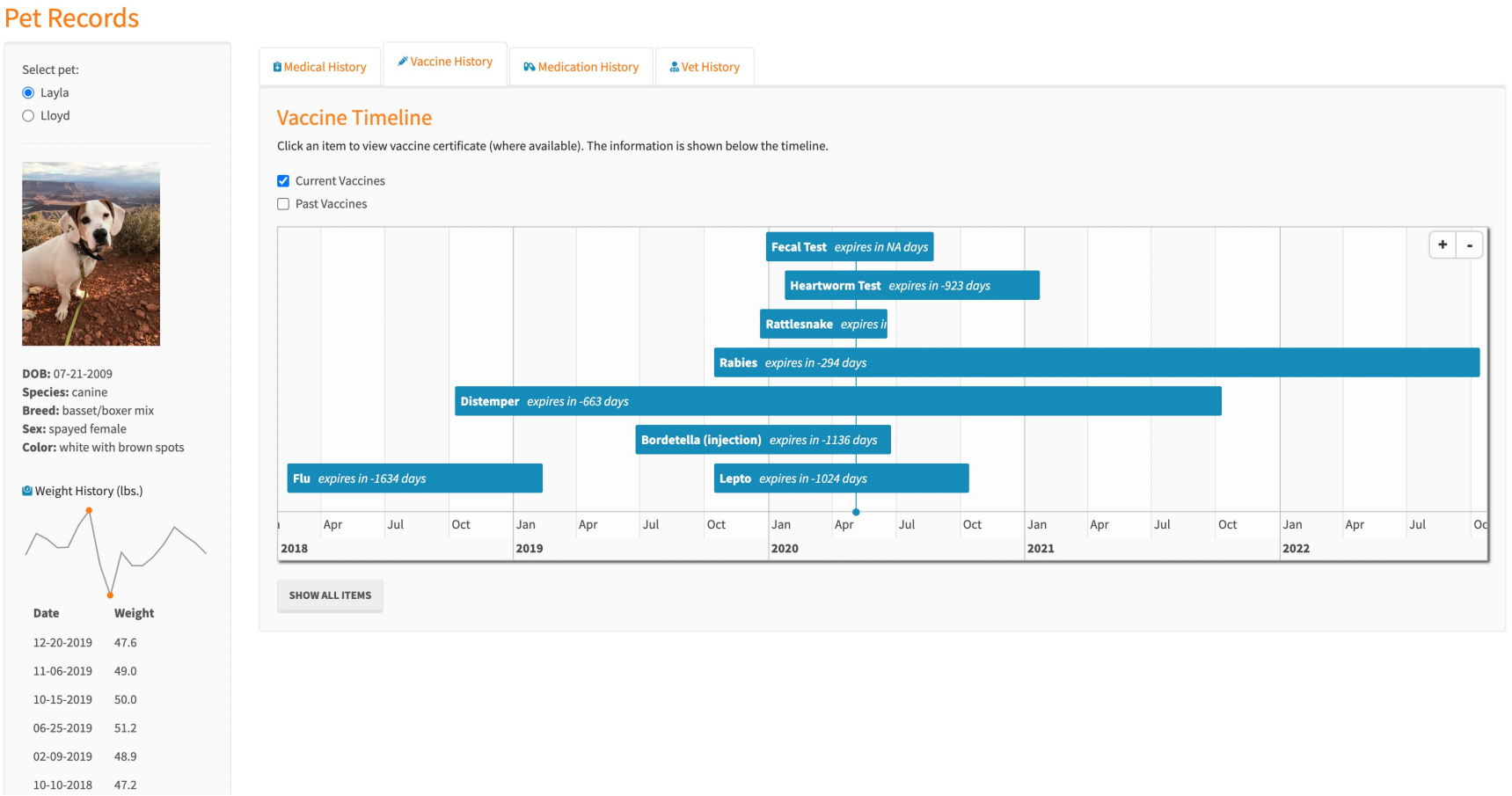
https://rstudio.github.io/shinydashboard/

Example apps

- Shiny for R Gallery
- 2019 Shiny Contest Winners
- 2020 Shiny Contest Winners
- 2021 Shiny Contest Winners



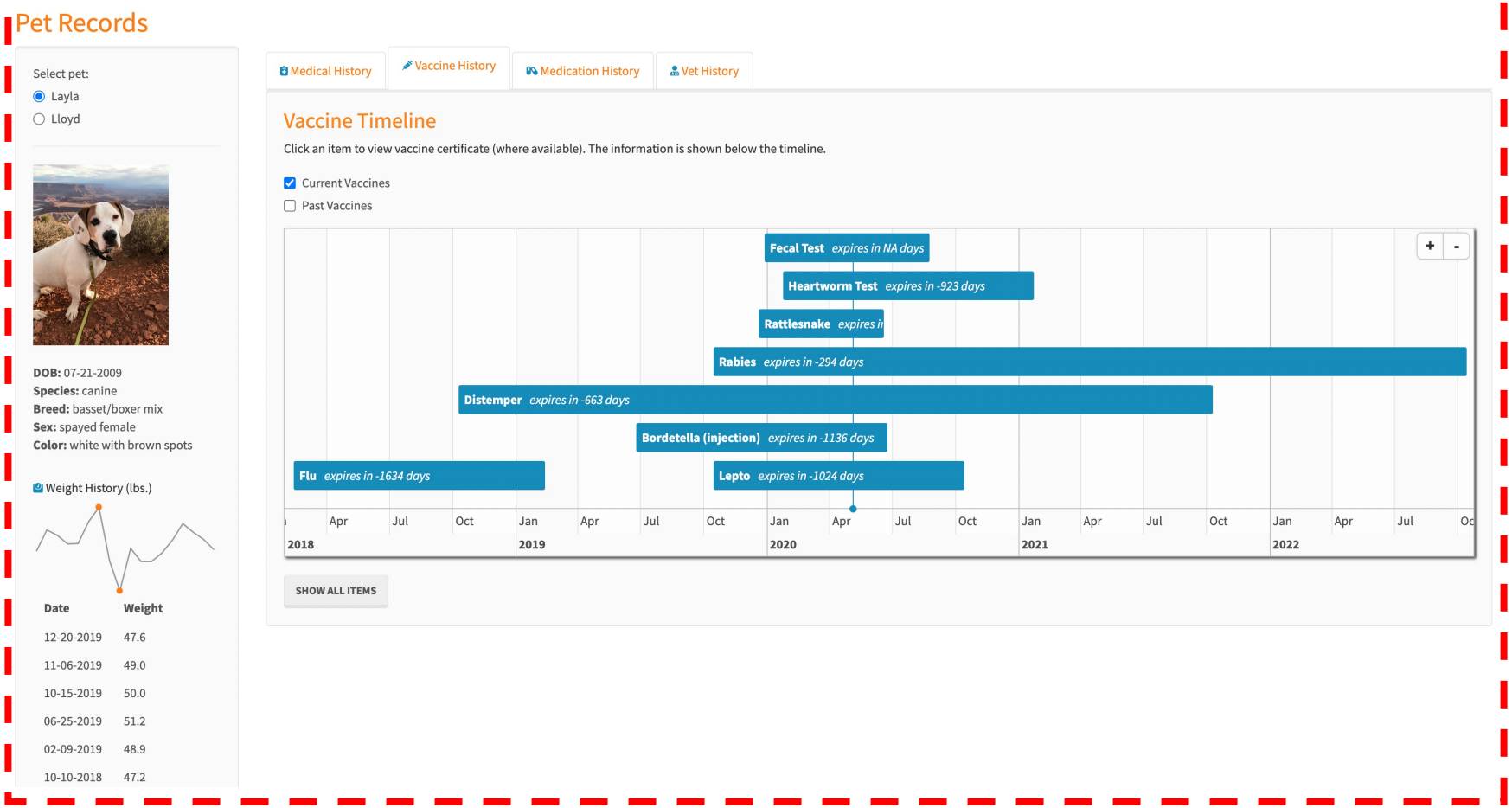
Guess the Layout: Pet Records



https://jennadallen.shinyapps.io/pet-records-app/

Pet Records Answer

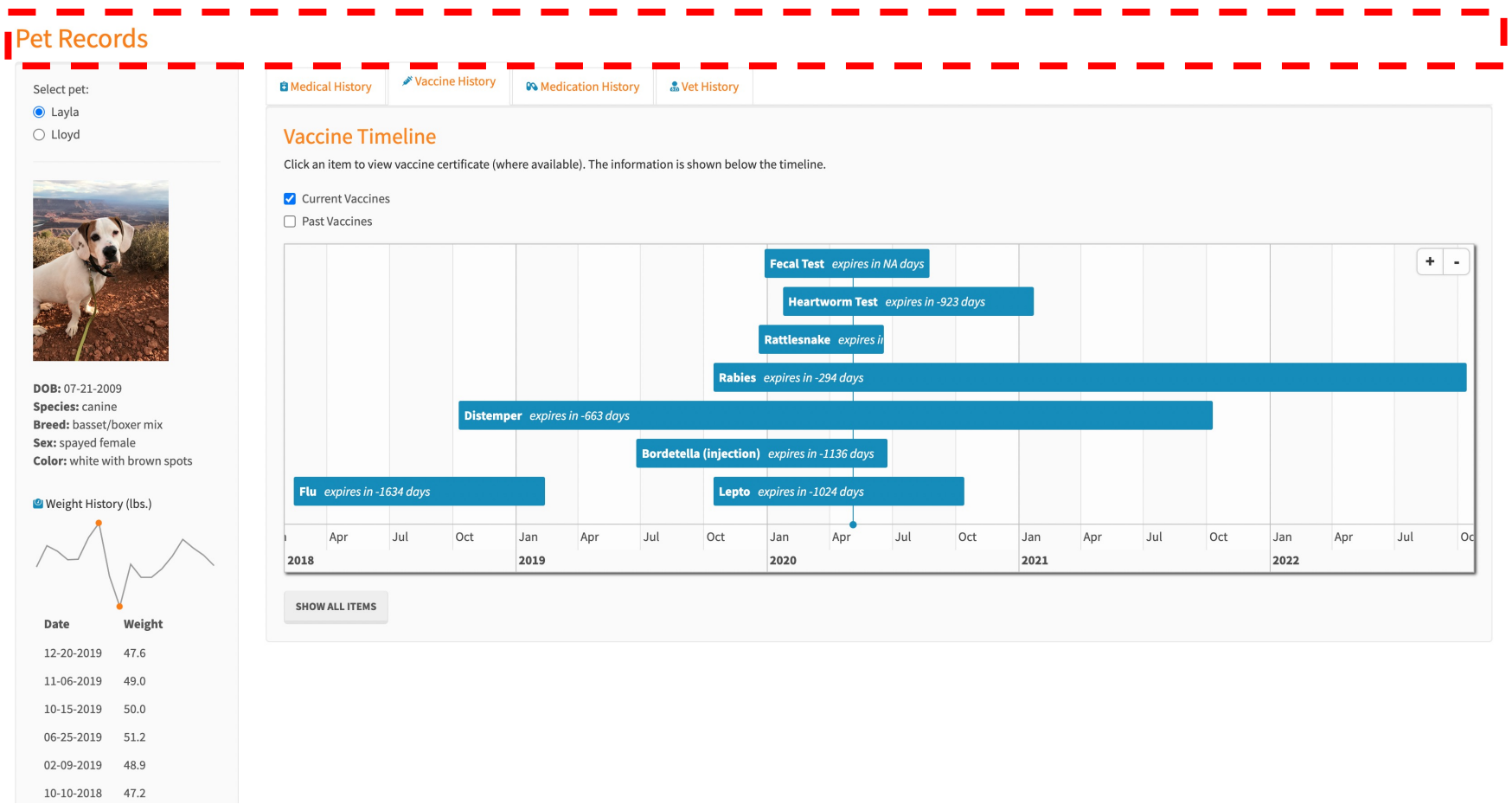
Fluid Page



https://github.com/jennaallen/dog_days

Pet Records Answer

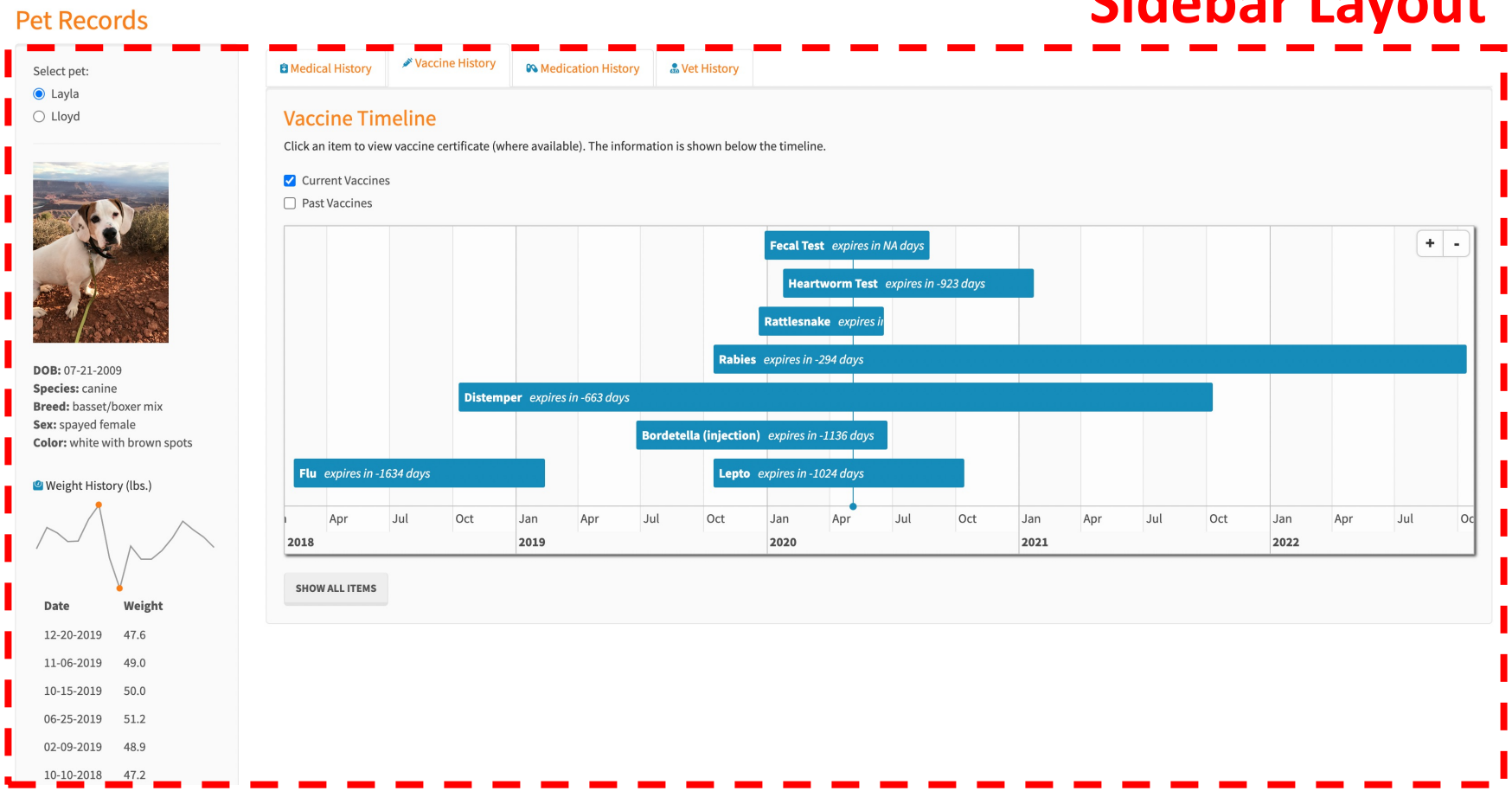
Title Panel



https://github.com/jennaallen/dog_days

Pet Records Answer

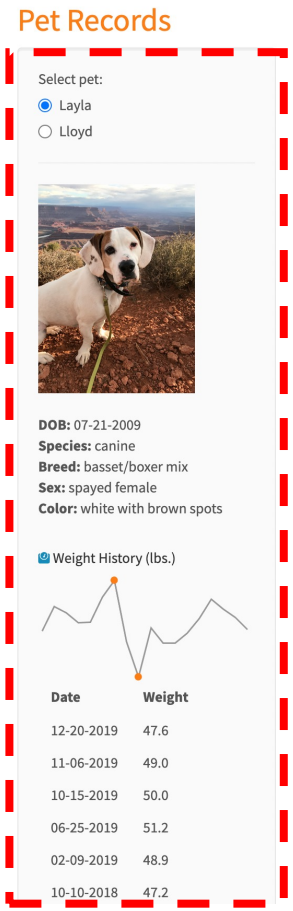
Sidebar Layout

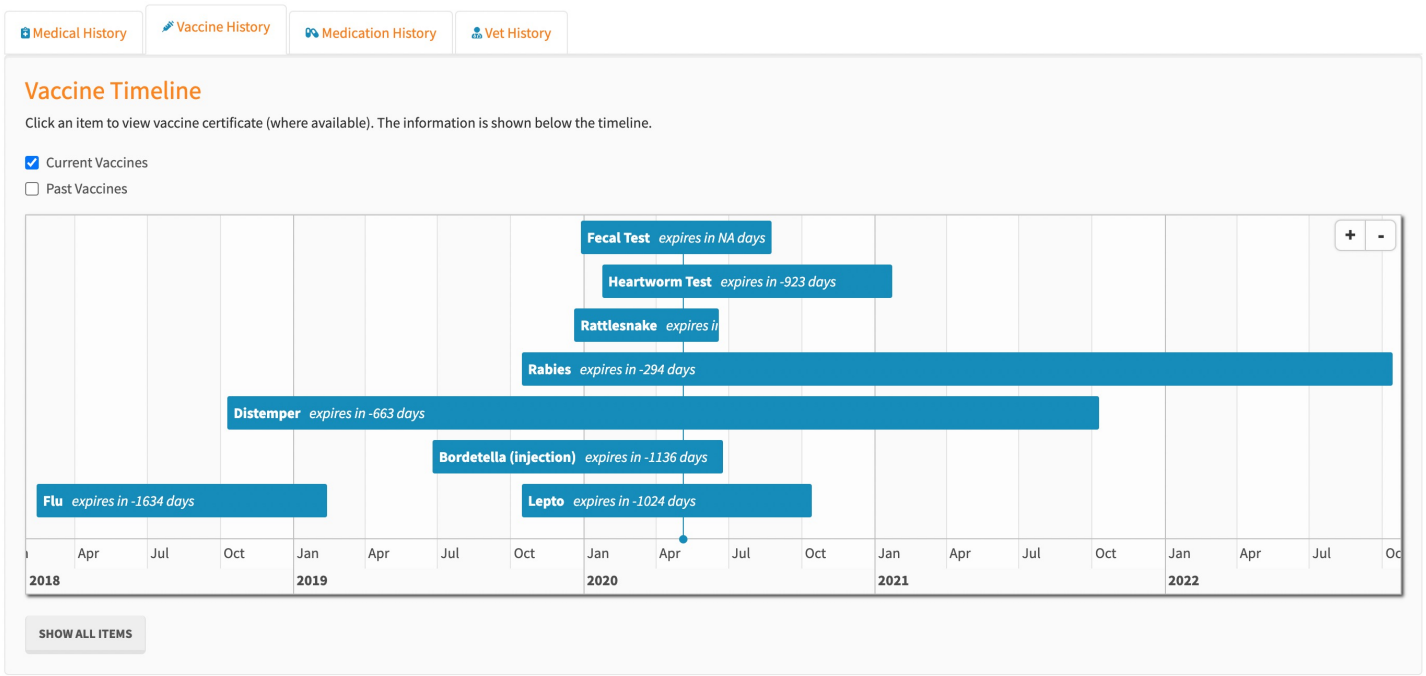


https://github.com/jennaallen/dog_days

Pet Records Answer

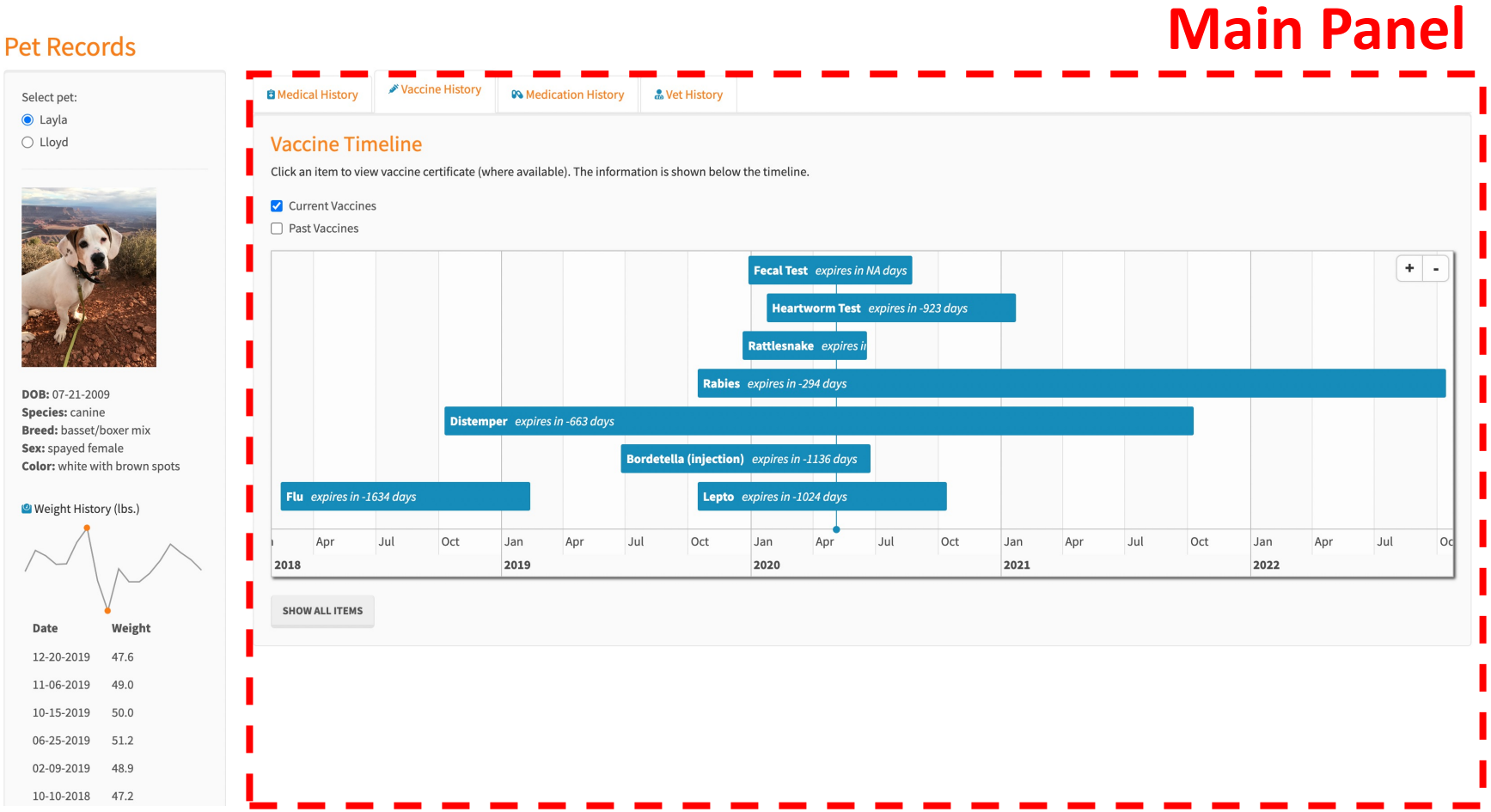
Sidebar Panel





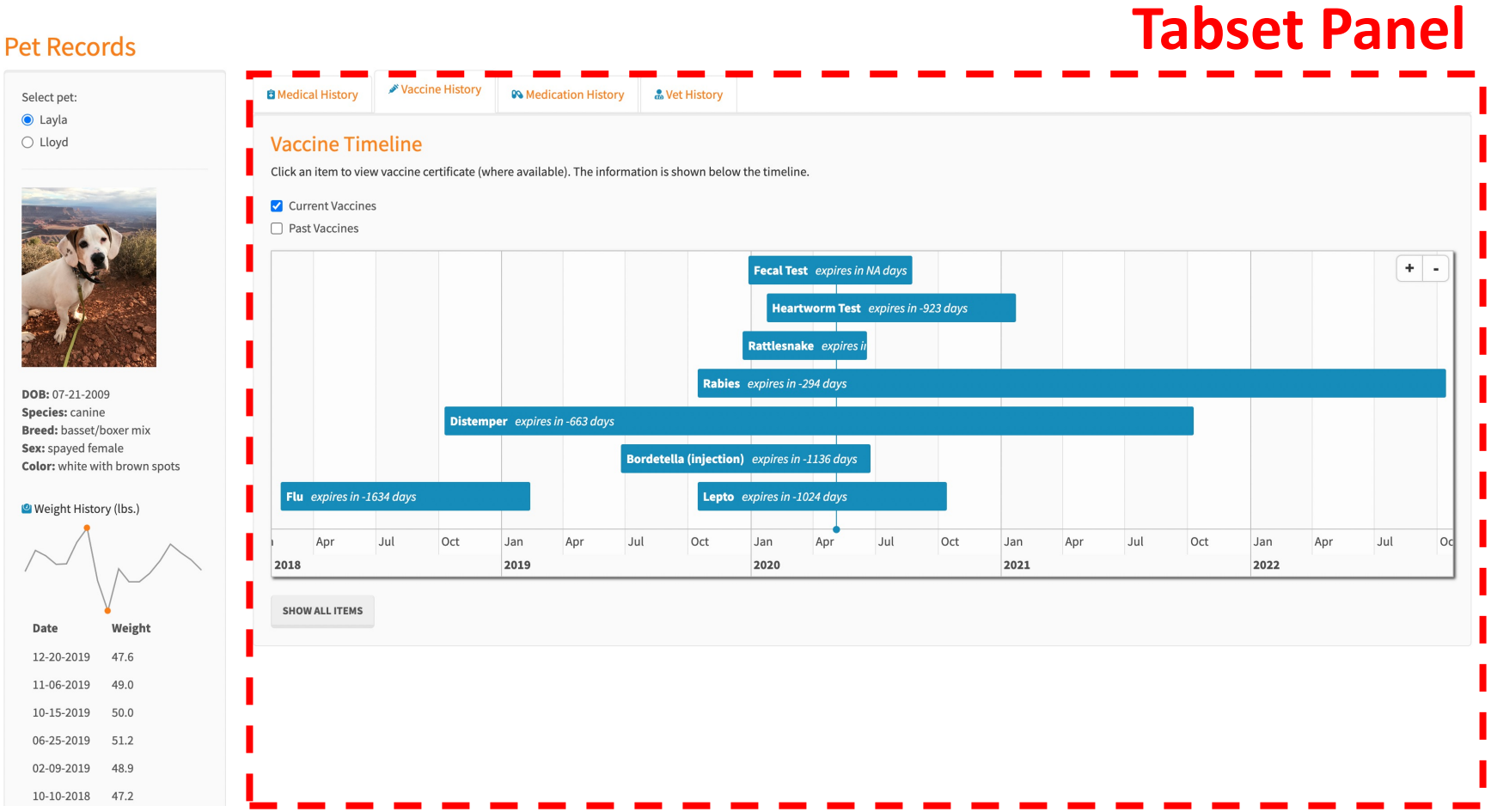
https://github.com/jennaallen/dog_days

Pet Records Answer



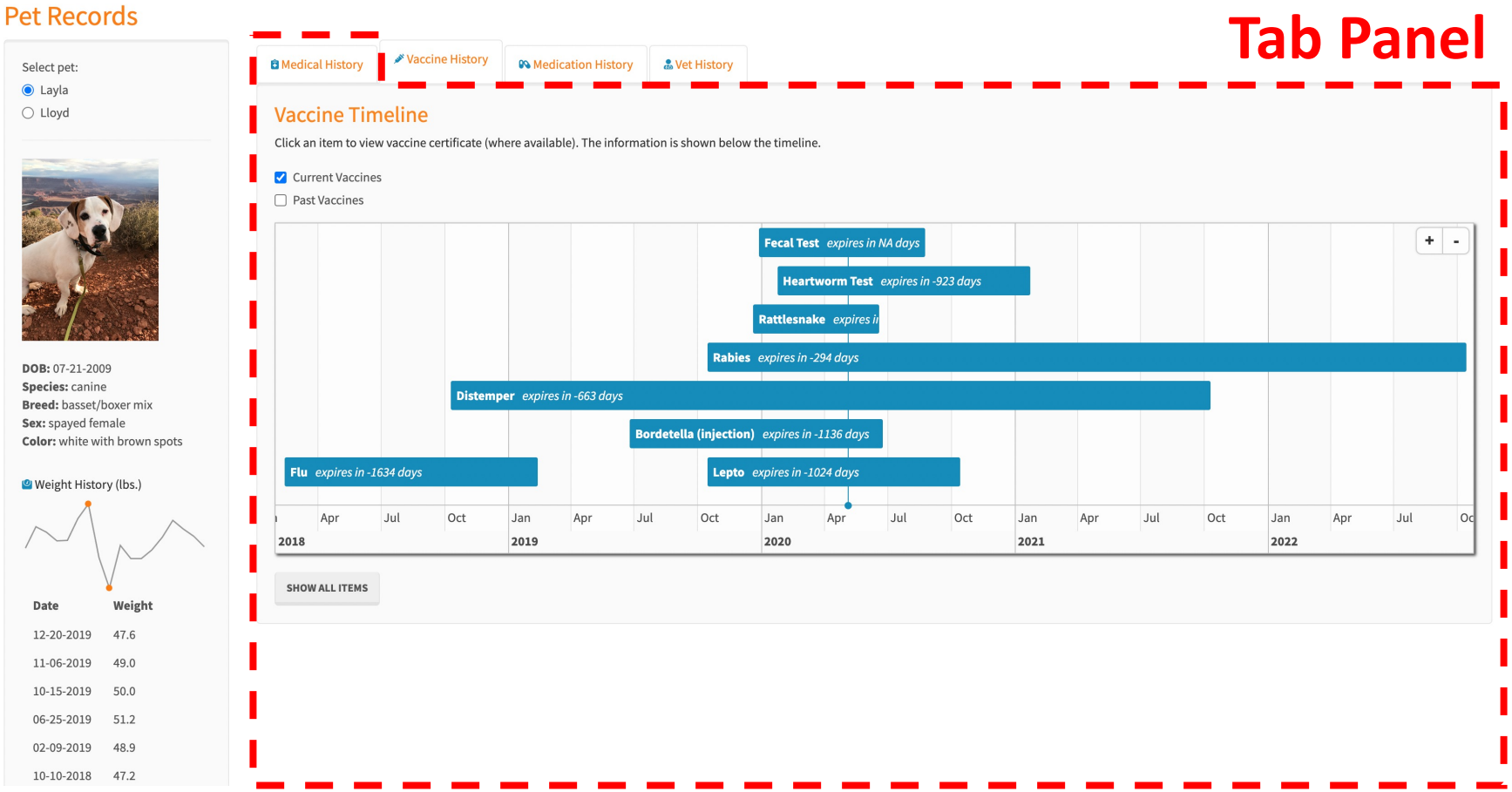
https://github.com/jennaallen/dog_days

Pet Records Answer



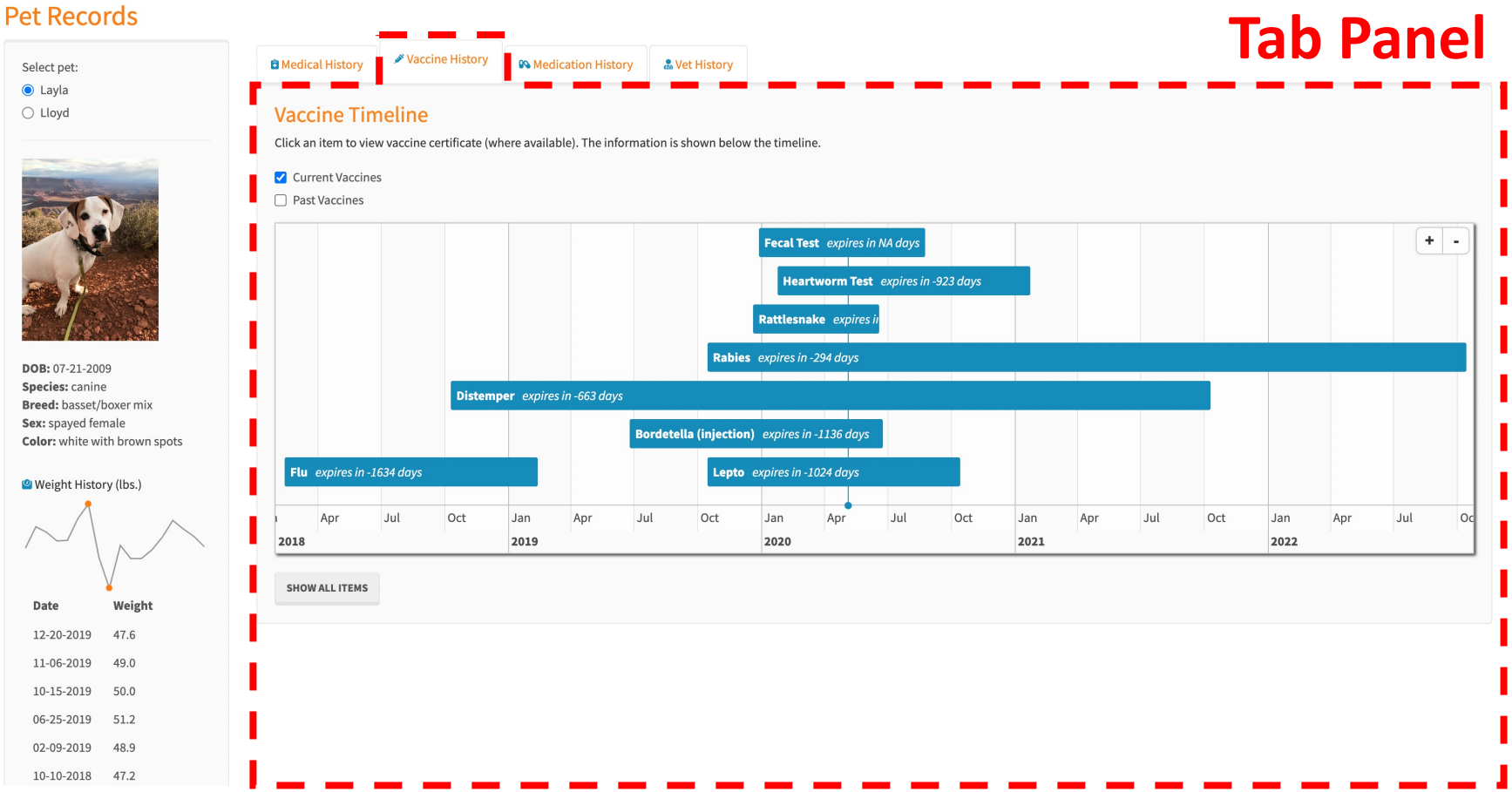
https://github.com/jennaallen/dog_days

Pet Records Answer



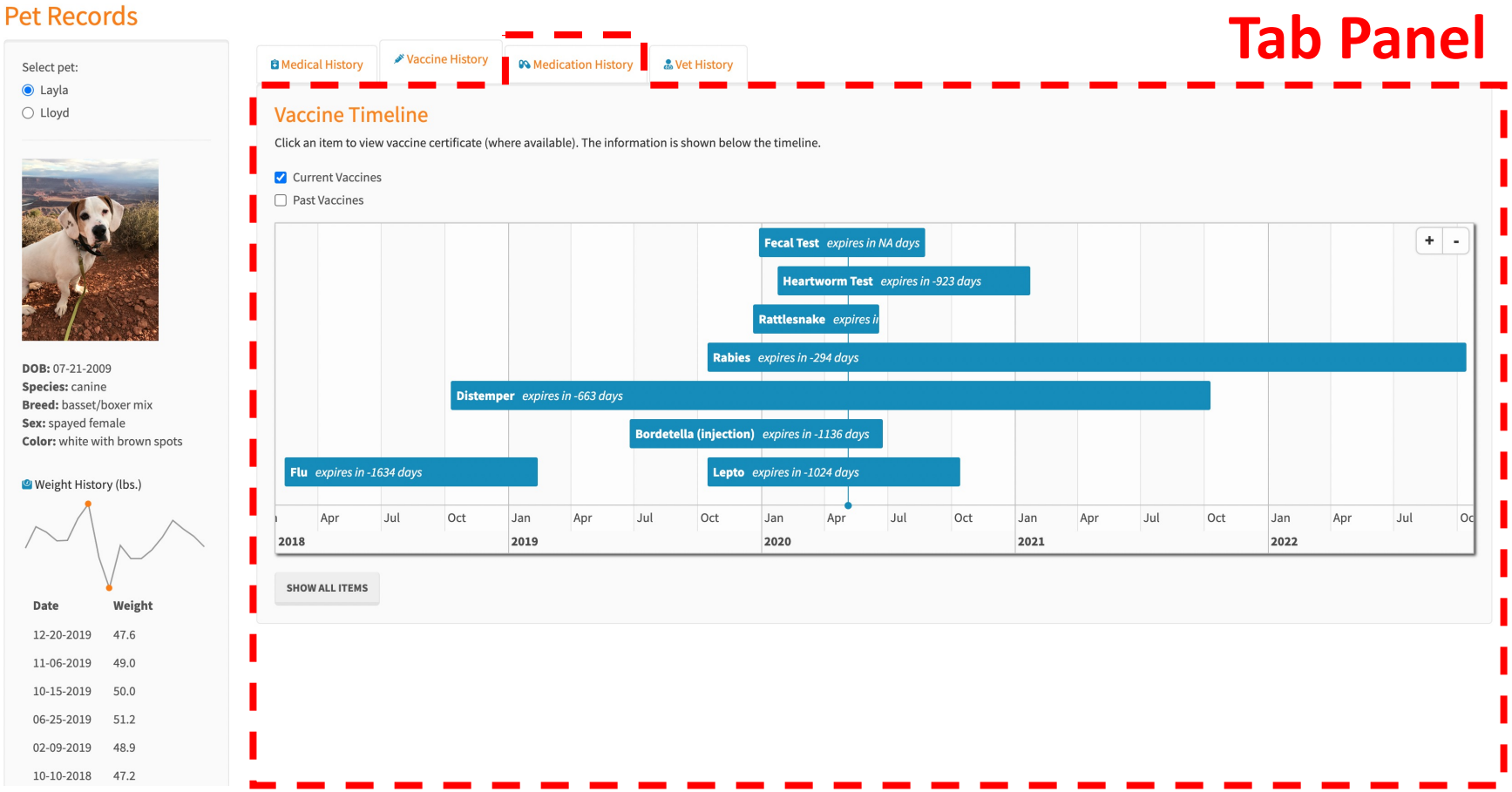
https://github.com/jennaallen/dog_days

Pet Records Answer



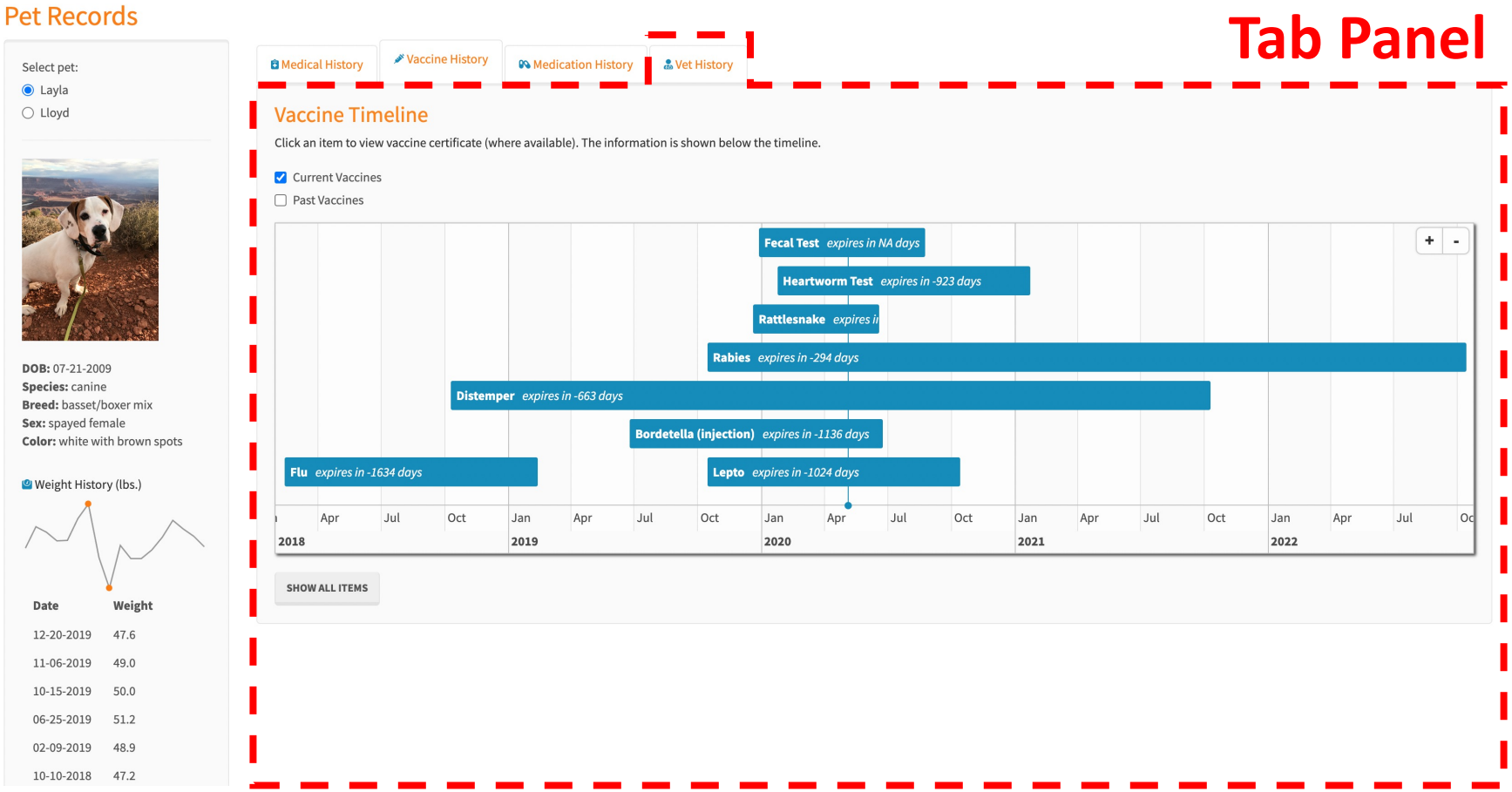
https://github.com/jennaallen/dog_days

Pet Records Answer



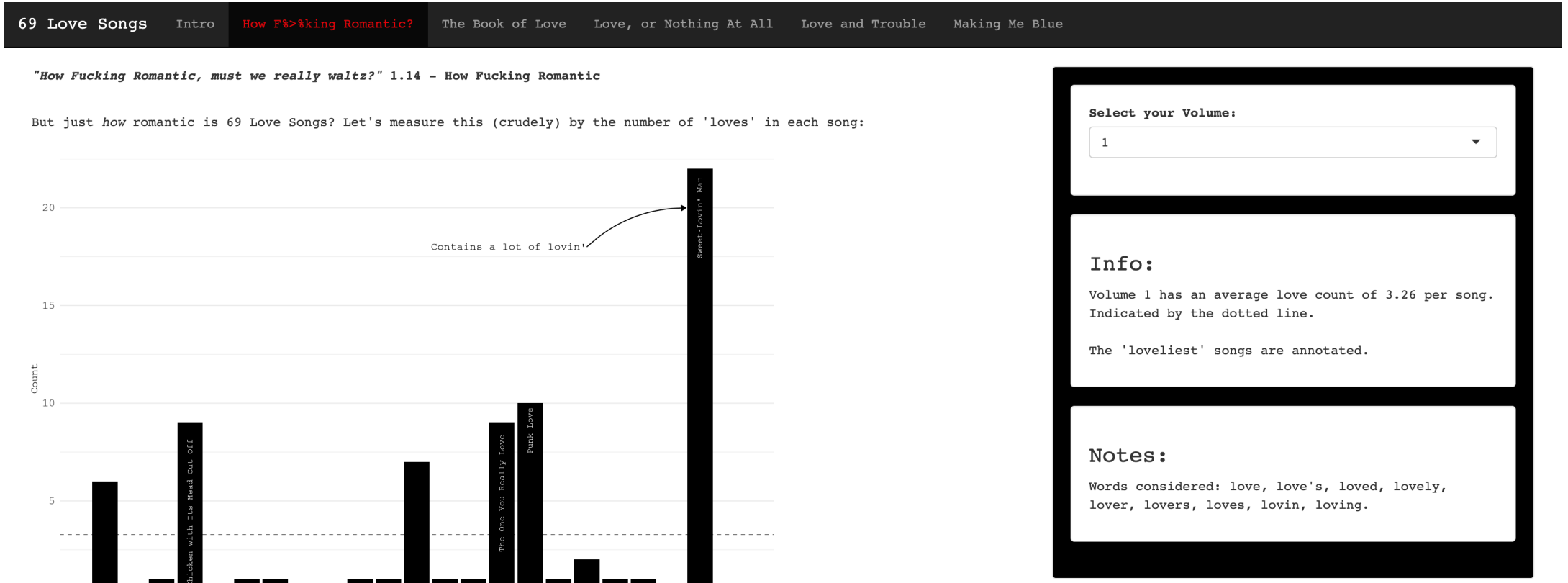
https://github.com/jennaallen/dog_days

Pet Records Answer



https://github.com/jennaallen/dog_days

Guess the Layout: 69 Love Songs



https://committedtotape.shinyapps.io/sixtyninelovesongs/

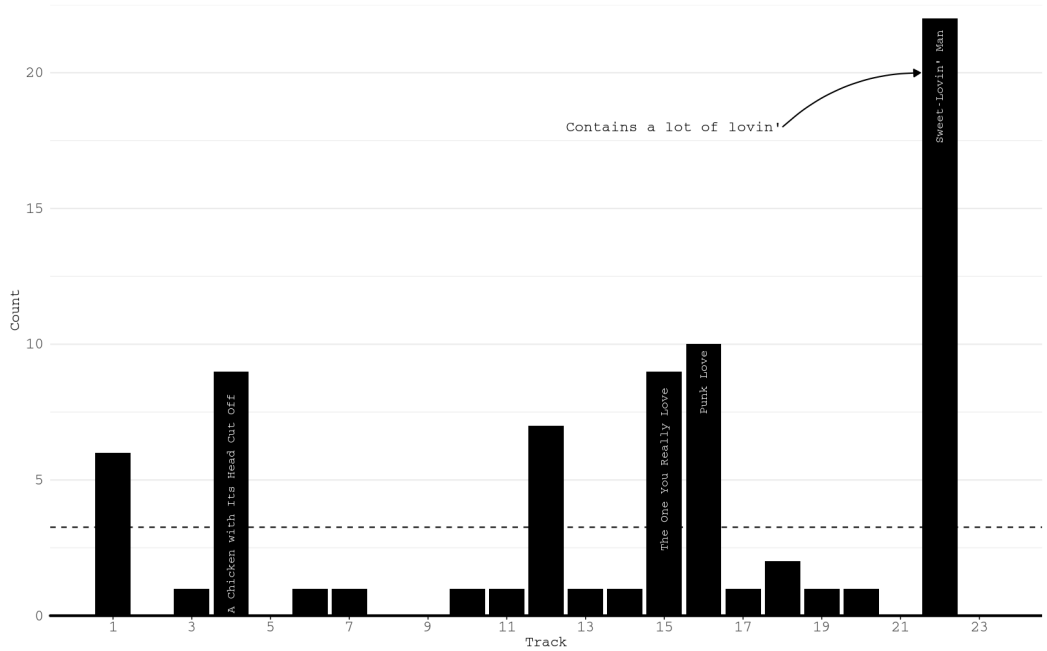
69 Love Songs Answer

Navbar Page

69 Love Songs | Intro | How F%>%king Romantic? | The Book of Love | Love, or Nothing At All | Love and Trouble | Making Me Blue

***"How Fucking Romantic, must we really waltz?"* 1.14 - How Fucking Romantic**

But just *how* romantic is 69 Love Songs? Let's measure this (crudely) by the number of 'loves' in each song:



Select your Volume:

1

Info:

Volume 1 has an average love count of 3.26 per song. Indicated by the dotted line.

The 'loveliest' songs are annotated.

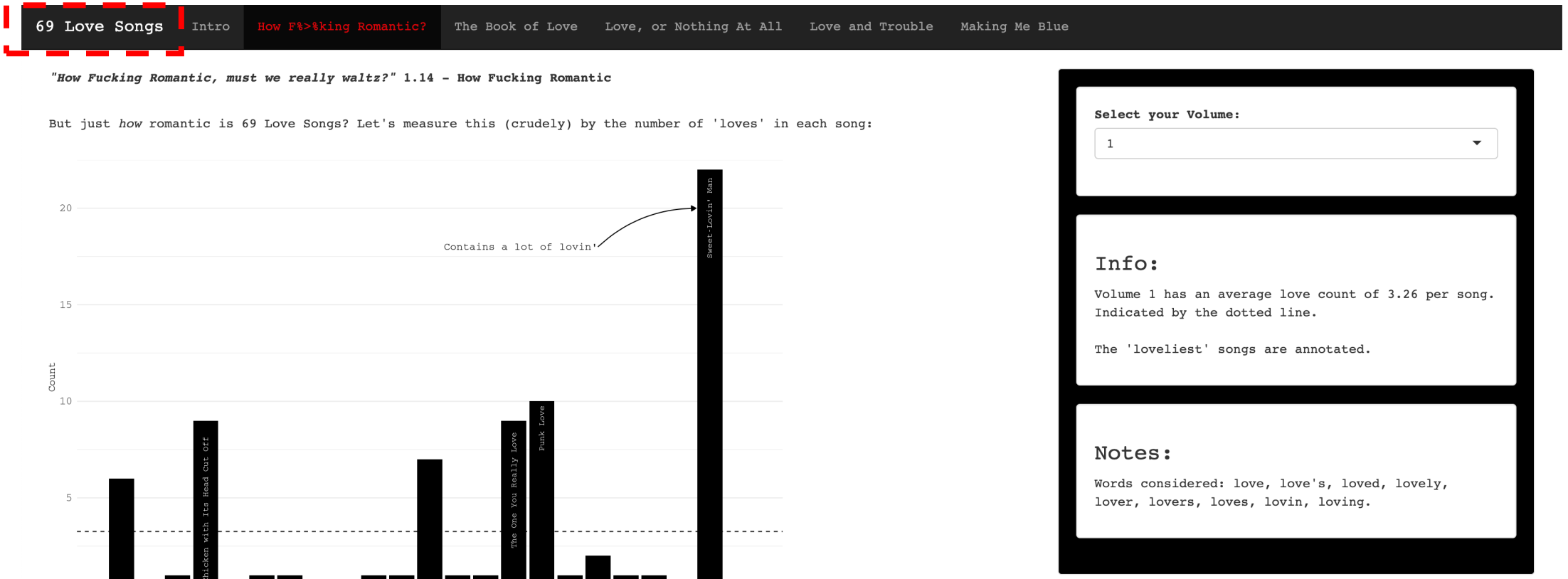
Notes:

Words considered: love, love's, loved, lovely, lover, lovers, loves, lovin, loving.

https://committedtotape.shinyapps.io/sixtyninelovesongs/

69 Love Songs Answer

Navbar Page Title



https://committedtotape.shinyapps.io/sixtyninelovesongs/

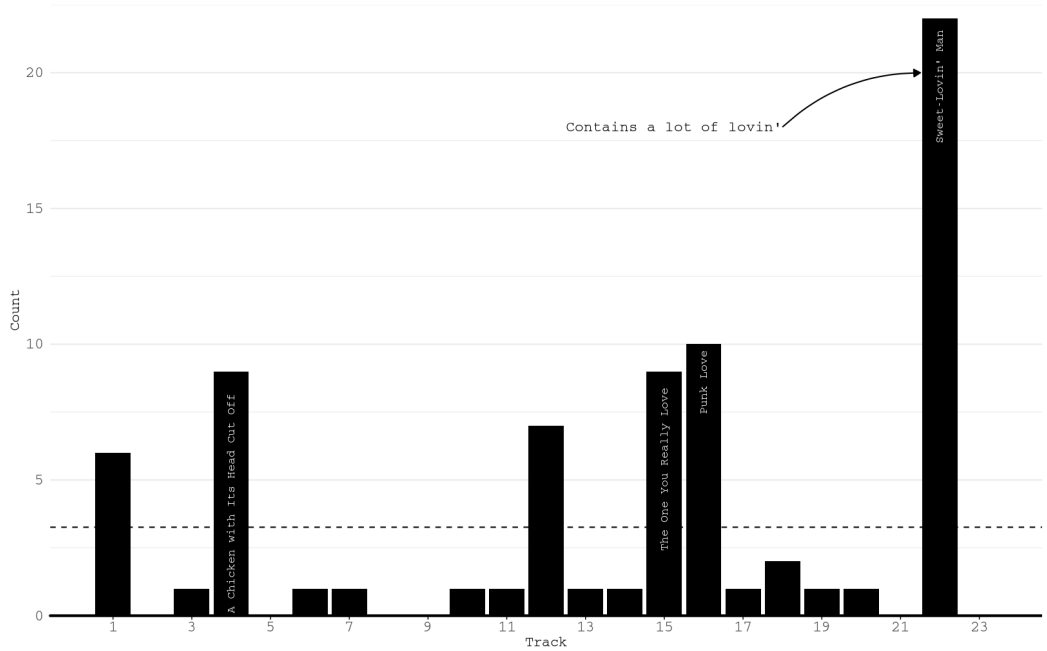
69 Love Songs Answer

Tab Panel

69 Love Songs | Intro | How F%>%king Romantic? | The Book of Love | Love, or Nothing At All | Love and Trouble | Making Me Blue

"How Fucking Romantic, must we really waltz?" **1.14 - How Fucking Romantic**

But just *how* romantic is 69 Love Songs? Let's measure this (crudely) by the number of 'loves' in each song:



Select your Volume:

1

Info:

Volume 1 has an average love count of 3.26 per song. Indicated by the dotted line.

The 'loveliest' songs are annotated.

Notes:

Words considered: love, love's, loved, lovely, lover, lovers, loves, lovin, loving.

https://committedtotape.shinyapps.io/sixtyninelovesongs/

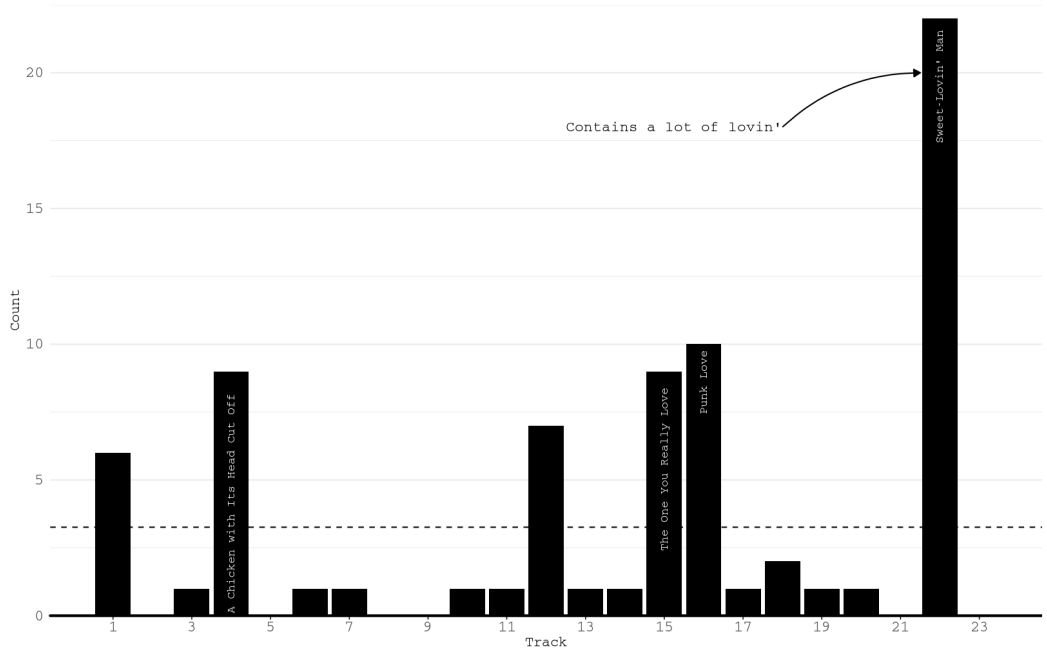
69 Love Songs Answer

Tab Panel

69 Love Songs | Intro | How F%>%king Romantic? | The Book of Love | Love, or Nothing At All | Love and Trouble | Making Me Blue

"How Fucking Romantic, must we really waltz?" **1.14 - How Fucking Romantic**

But just *how* romantic is 69 Love Songs? Let's measure this (crudely) by the number of 'loves' in each song:



Select your Volume:

1

Info:

Volume 1 has an average love count of 3.26 per song. Indicated by the dotted line.

The 'loveliest' songs are annotated.

Notes:

Words considered: love, love's, loved, lovely, lover, lovers, loves, lovin, loving.

https://committedtotape.shinyapps.io/sixtyninelovesongs/

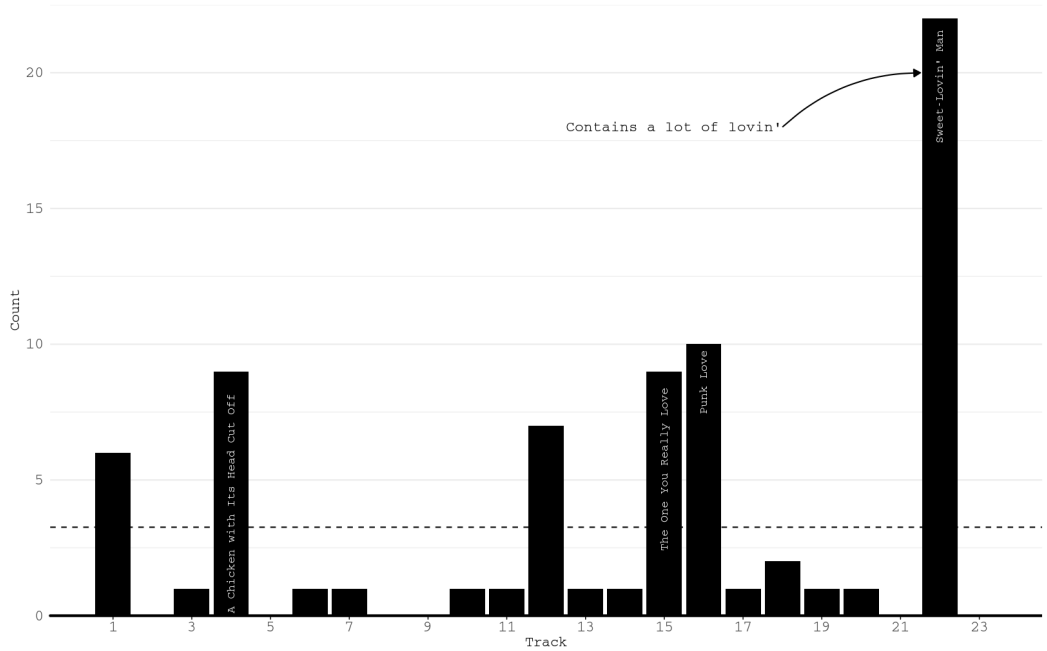
69 Love Songs Answer

Fluid Page

69 Love Songs | Intro | How F%>%king Romantic? | The Book of Love | Love, or Nothing At All | Love and Trouble | Making Me Blue

"How Fucking Romantic, must we really waltz?" **1.14 - How Fucking Romantic**

But just *how* romantic is 69 Love Songs? Let's measure this (crudely) by the number of 'loves' in each song:



Select your Volume:

1

Info:

Volume 1 has an average love count of 3.26 per song. Indicated by the dotted line.

The 'loveliest' songs are annotated.

Notes:

Words considered: love, love's, loved, lovely, lover, lovers, loves, lovin, loving.

https://committedtotape.shinyapps.io/sixtyninelovesongs/

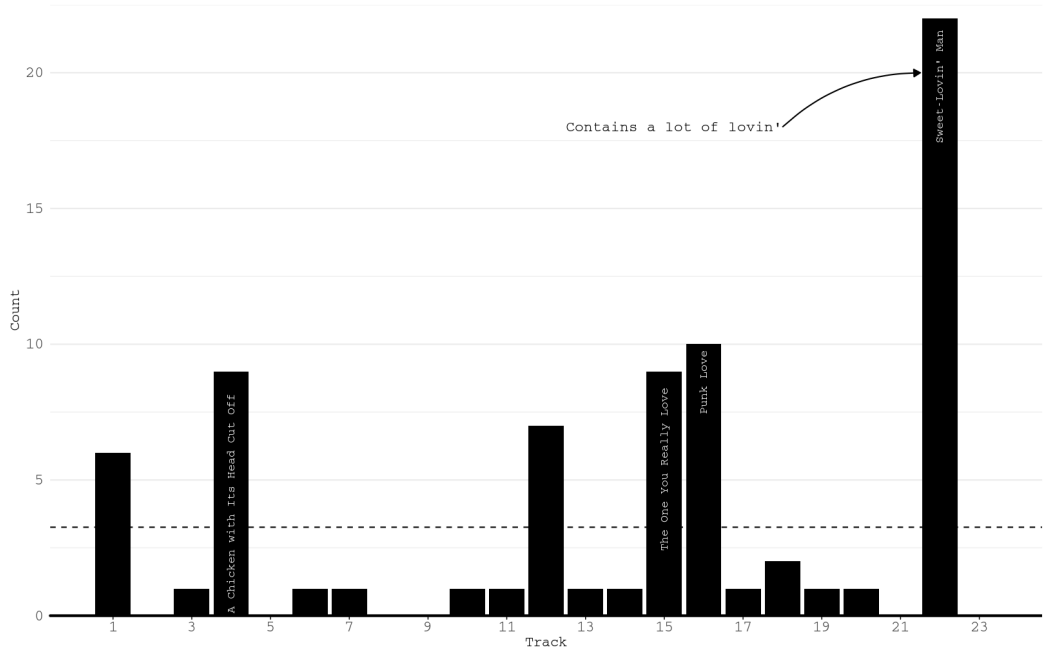
69 Love Songs Answer

Sidebar Layout

69 Love Songs | Intro | How F%>%king Romantic? | The Book of Love | Love, or Nothing At All | Love and Trouble | Making Me Blue

"How Fucking Romantic, must we really waltz?" **1.14 - How Fucking Romantic**

But just *how* romantic is 69 Love Songs? Let's measure this (crudely) by the number of 'loves' in each song:



Select your Volume:

1

Info:

Volume 1 has an average love count of 3.26 per song. Indicated by the dotted line.

The 'loveliest' songs are annotated.

Notes:

Words considered: love, love's, loved, lovely, lover, lovers, loves, lovin, loving.

https://committedtotape.shinyapps.io/sixtyninelovesongs/

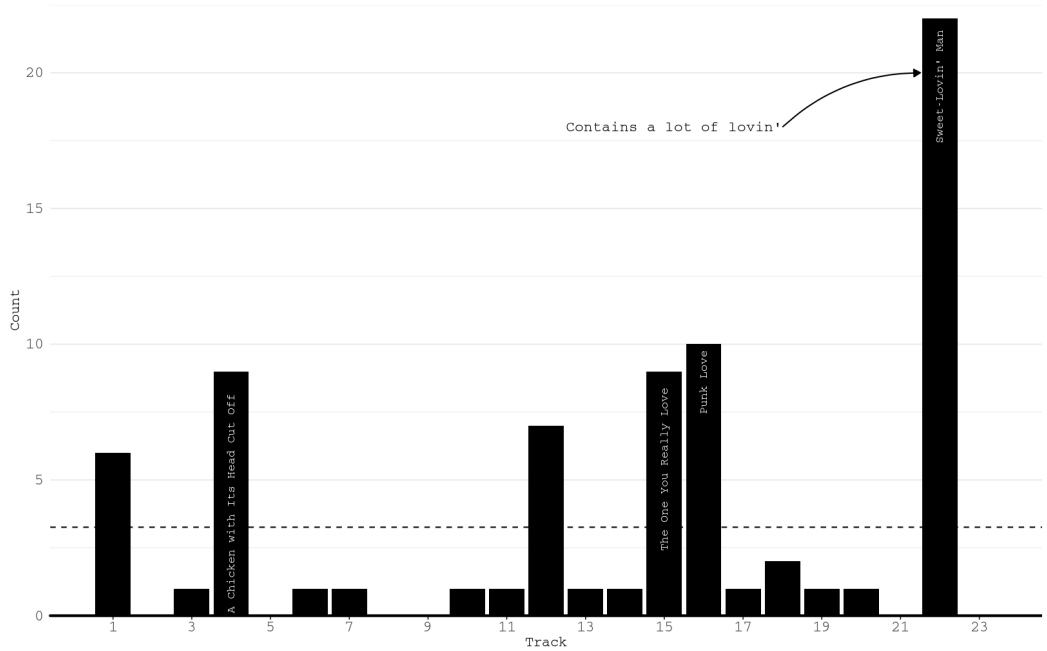
69 Love Songs Answer

Sidebar Panel (position = “right”)

69 Love Songs | Intro | How F%>%king Romantic? | The Book of Love | Love, or Nothing At All | Love and Trouble | Making Me Blue

"How Fucking Romantic, must we really waltz?" **1.14 - How Fucking Romantic**

But just *how* romantic is 69 Love Songs? Let's measure this (crudely) by the number of 'loves' in each song:



Select your Volume:

1

Info:

Volume 1 has an average love count of 3.26 per song. Indicated by the dotted line.

The 'loveliest' songs are annotated.

Notes:

Words considered: love, love's, loved, lovely, lover, lovers, loves, lovin, loving.

https://committedtotape.shinyapps.io/sixtyninelovesongs/

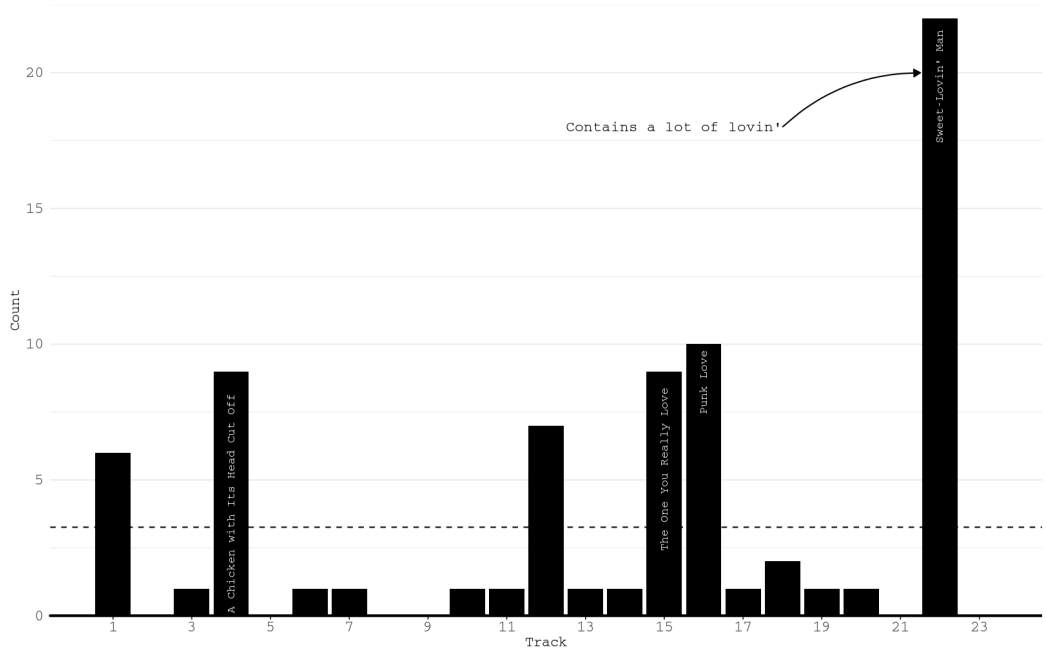
69 Love Songs Answer

Well Panels

69 Love Songs | Intro | How F%>%king Romantic? | The Book of Love | Love, or Nothing At All | Love and Trouble | Making Me Blue

***"How Fucking Romantic, must we really waltz?"* 1.14 - How Fucking Romantic**

But just *how* romantic is 69 Love Songs? Let's measure this (crudely) by the number of 'loves' in each song:



Select your Volume:

1

Info:

Volume 1 has an average love count of 3.26 per song. Indicated by the dotted line.

The 'loveliest' songs are annotated.

Notes:

Words considered: love, love's, loved, lovely, lover, lovers, loves, lovin, loving.

https://committedtotape.shinyapps.io/sixtyninelovesongs/

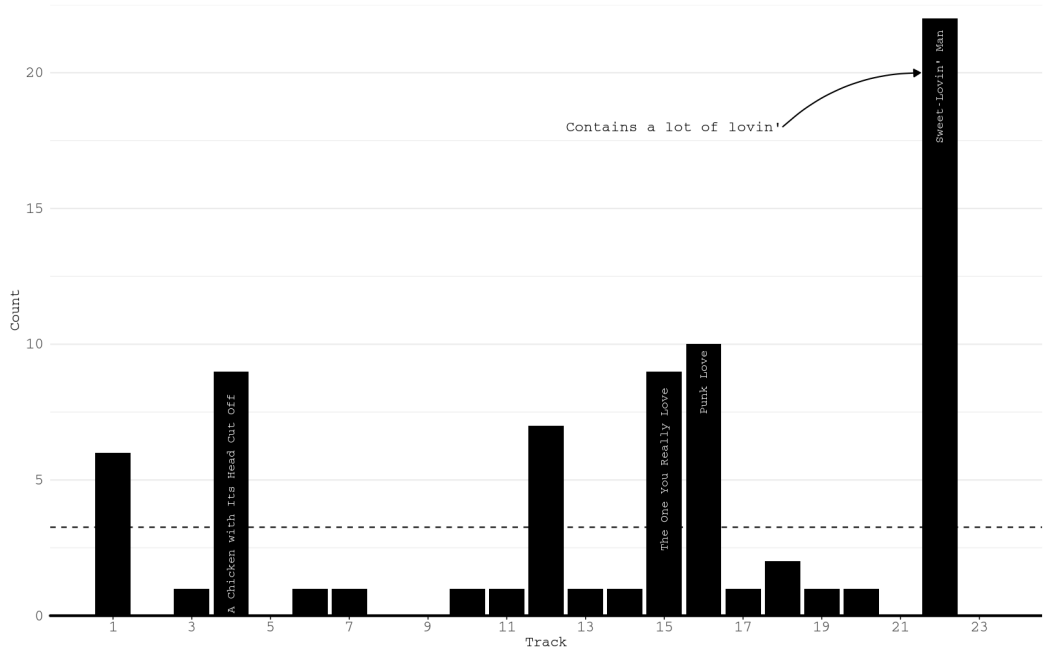
69 Love Songs Answer

Main Panel

69 Love Songs | Intro | How F%>%king Romantic? | The Book of Love | Love, or Nothing At All | Love and Trouble | Making Me Blue

"How Fucking Romantic, must we really waltz?" **1.14 - How Fucking Romantic**

But just *how* romantic is 69 Love Songs? Let's measure this (crudely) by the number of 'loves' in each song:



Select your Volume:

1

Info:

Volume 1 has an average love count of 3.26 per song. Indicated by the dotted line.

The 'loveliest' songs are annotated.

Notes:

Words considered: love, love's, loved, lovely, lover, lovers, loves, lovin, loving.

https://committedtotape.shinyapps.io/sixtyninelovesongs/

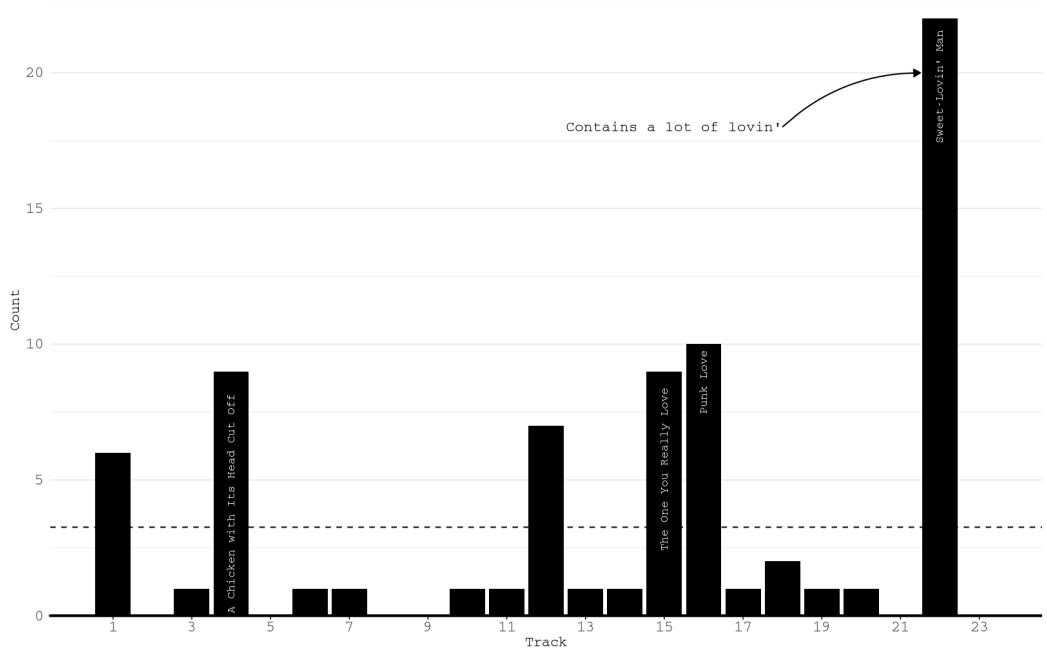
69 Love Songs Answer

Tab Panels

69 Love Songs | Intro | How F%>%king Romantic? | The Book of Love | Love, or Nothing At All | Love and Trouble | Making Me Blue

"How Fucking Romantic, must we really waltz?" **1.14 - How Fucking Romantic**

But just *how* romantic is 69 Love Songs? Let's measure this (crudely) by the number of 'loves' in each song:



Select your Volume:

1

Info:

Volume 1 has an average love count of 3.26 per song. Indicated by the dotted line.

The 'loveliest' songs are annotated.

Notes:

Words considered: love, love's, loved, lovely, lover, lovers, loves, lovin, loving.

https://committedtotape.shinyapps.io/sixtyninelovesongs/

Morning Break

Adding visual components or “outputs”

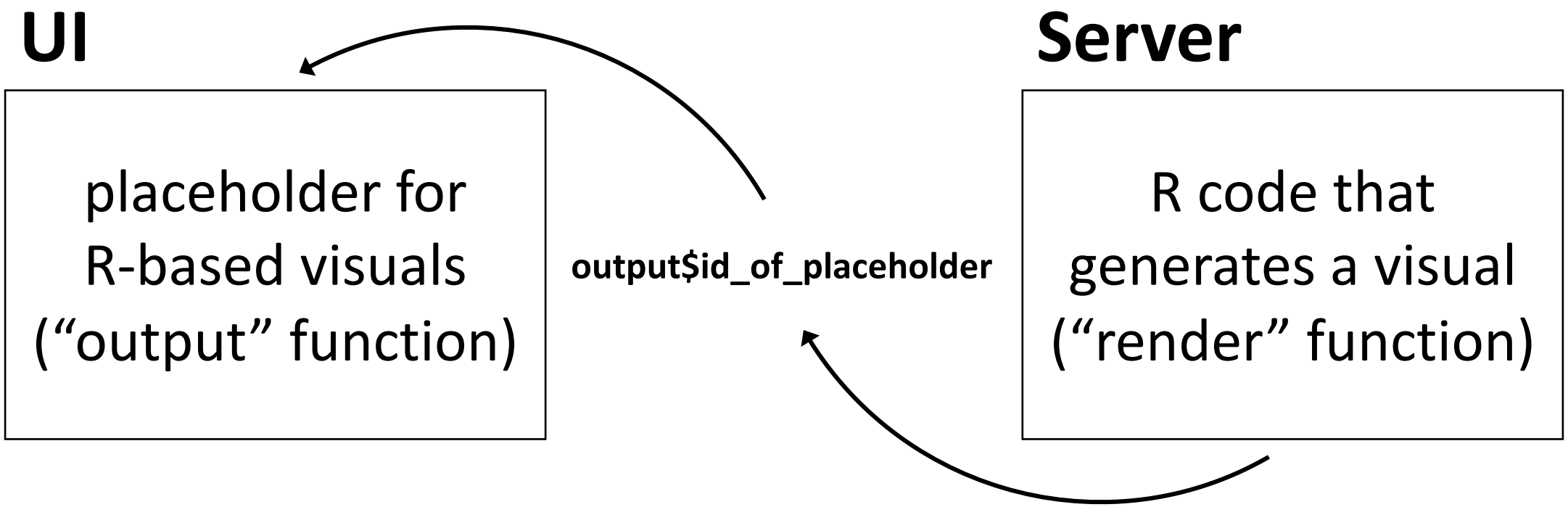
How do we use R to create visual content?

So far, we have only been displaying static content, using functions inspired by HTML tags.

Our layout containers don't accept normal R code.
(Why do you think this is?)

Where does the R code go? How does the result of the R code get displayed in the web app?

UI code specifies a placeholder to indicate where the R output should go and gives the output an identifier.



Server code saves R output to an object called "output", using the same ID listed in the placeholder.

Simple outputs and renders

Render objects with same names and types as the ones listed in output placeholders

UI:

```
textOutput(outputId = "text1")
```

Server:

```
output$text1 <- renderText({
                  "some text"
                })
```

https://shiny.posit.co/r/getstarted/shiny-basics/lesson4/

Output-Render Pairs

UI (output)	Server (render)
htmlOutput	renderUI
imageOutput	renderImage
plotOutput	renderPlot
tableOutput	renderTable
dataTableOutput	renderDataTable
textOutput	renderText
uiOutput	renderUI
verbatimTextOutput	renderPrint

https://shiny.posit.co/r/getstarted/shiny-basics/lesson4/

Exercise 3: Display a Plot

- Create an app with a sidebar layout.
- Leave the sidebar empty for now.
- In the main panel, add a placeholder for a plot.
- In the server, add code for a plot.

Exercise 4: Testing Different Outputs

- Create an app with a navigation bar.
- Create five pages to test different outputs:
  - Image output
  - Table output
  - Verbatim text output
  - Text output
  - HTML output

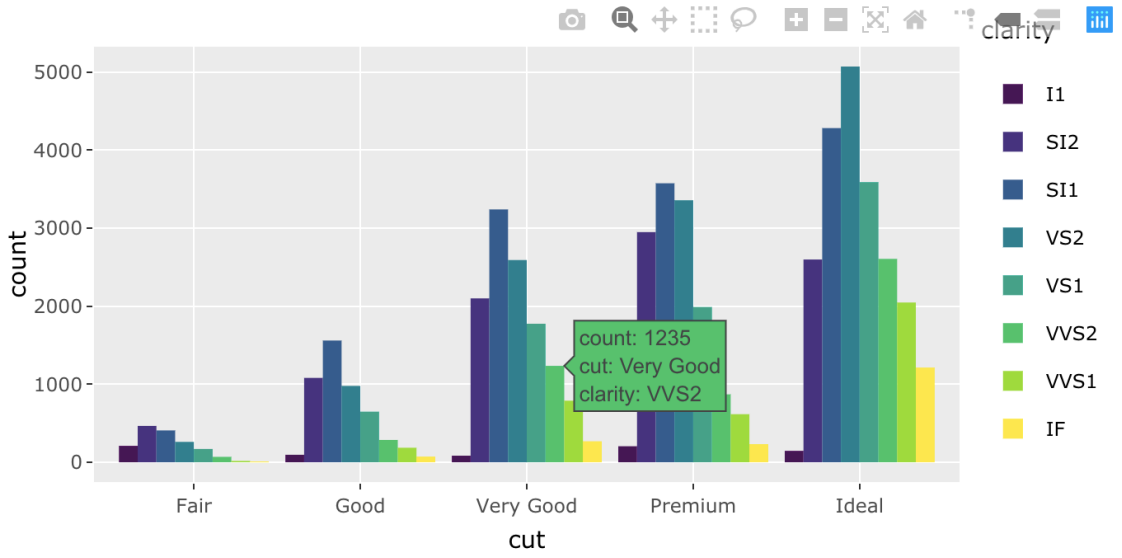
Interactive components

See “InteractiveOutputs” in examples folder

HTML widgets

- Some R packages can create interactive elements by outputting HTML/JavaScript code instead of normal R visuals
- These HTML widgets are useful for both R Markdown documents and Shiny apps, since both render HTML

http://www.htmlwidgets.org/



Show 5 entries Search:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5	3.6	1.4	0.2	setosa

Showing 1 to 5 of 150 entries

Previous 1 2 3 4 5 … 30 Next

Connecting inputs to outputs

How do we allow user feedback to change visual elements in the app?

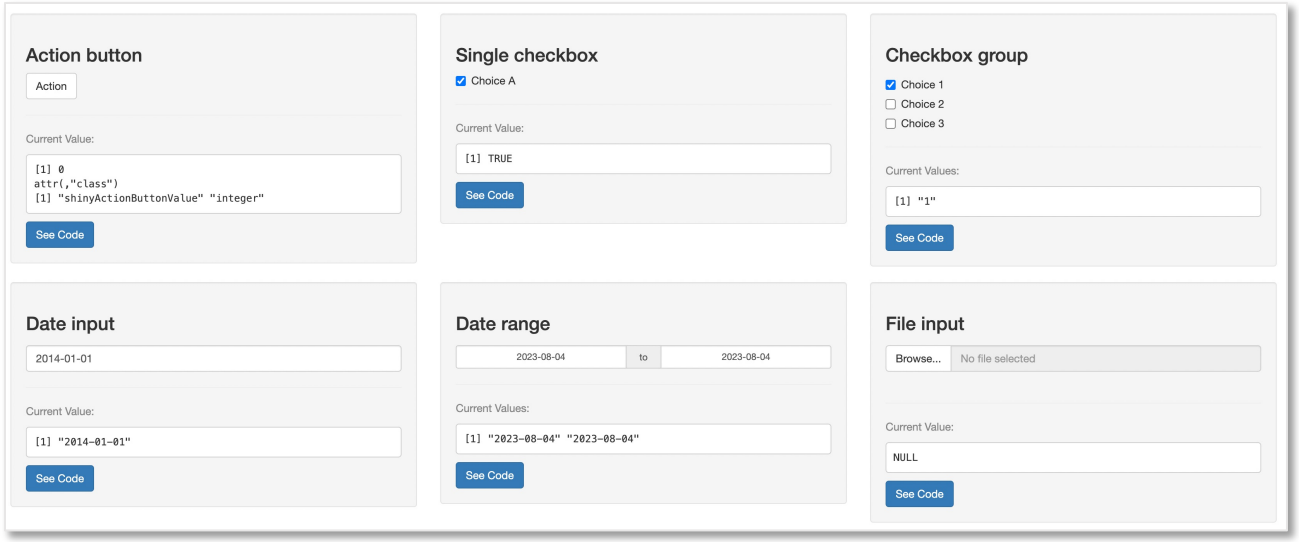
Now we have a way to inject the output of R code into our web app.

But if we are trying to help our user explore the data, we need to give them a way to change the display.

How can we design the app to accept user feedback? How can our R code respond to that feedback?

Input widgets

- Button
- Checkboxes
- Date, date range input
- File input
- Numeric input
- Radio buttons
- Drop-down (select) box
- Slider, slider range
- Text input

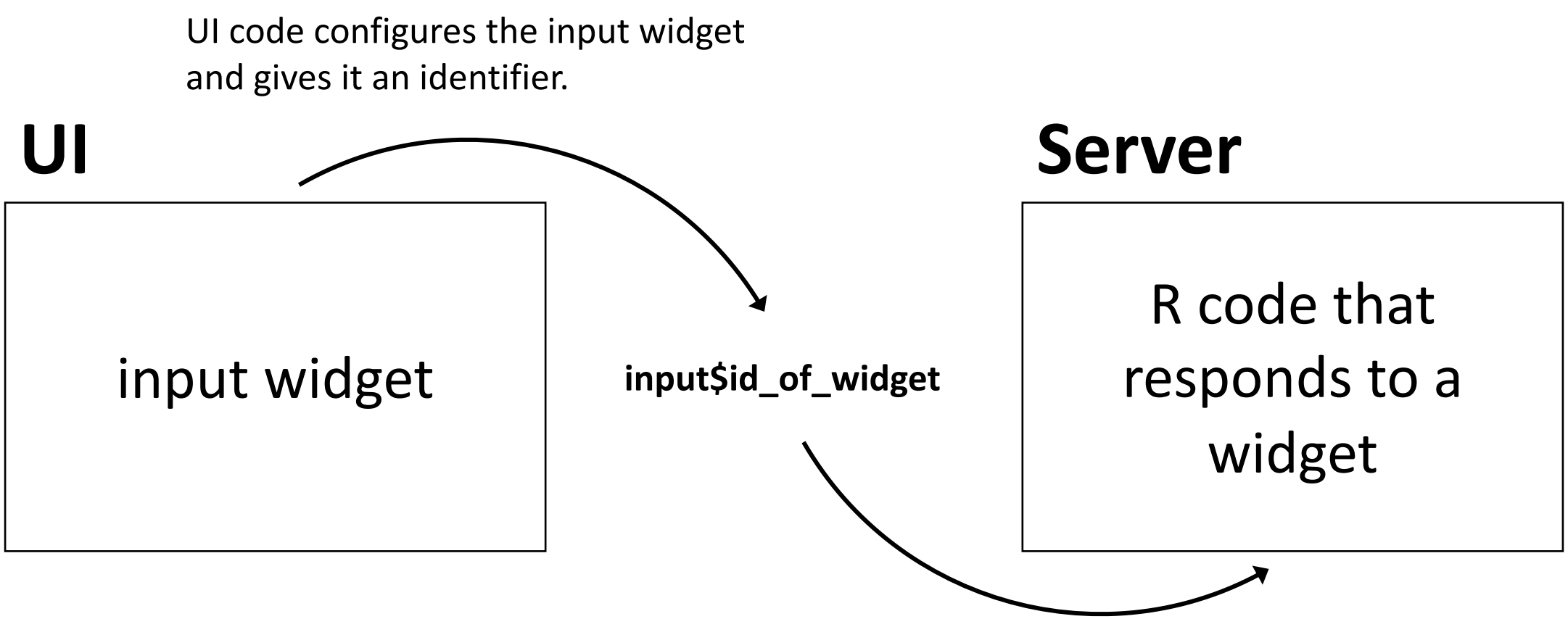


https://shiny.posit.co/r/getstarted/shiny-basics/lesson3/
https://shiny.posit.co/r/gallery/widgets/widget-gallery/

Anatomy of an input widget

- **inputId** for the widget (internal only)
- **label** (will be visible)
- Check documentation for other required arguments (e.g., selectInput requires choices)


```
selectInput(inputId = "selectBox",
            label = "Select box:",
            choices = c("Choice 1",
                        "Choice 2”,
                        "Choice 3")
```


UI code configures the input widget
and gives it an identifier.
UI
Server
input widget
input$id_of_widget
R code that
responds to a
widget
Server code watches the object called
“input”, using the same ID listed in the UI.
R code is rerun when the input changes.

Simple structure of inputs, outputs, and renders

- Render objects with same names and types as the ones listed in output placeholders
- Calls to input objects, using the same names as the control widgets

UI:

```
sliderInput(inputId = "slider1",
            ...)

textOutput(outputId = "text1")
```

Server:

```
output$text1 <- renderText({
                  input$slider1
                })
```

https://shiny.posit.co/r/getstarted/shiny-basics/lesson4/

Exercise 5: Basic Input

- Create an app with a sidebar layout.
- In the sidebar, add a drop-down menu with options 1, 2, and 3.
- In the main panel, add a placeholder for an image.
- In the server, use R code to render a different image depending on which option is selected in the drop-down.

Lunch break

Updating R code based on user feedback: Reactivity

Reactivity

Writing R code for a Shiny app is a bit different from normal Shiny code.

We've already seen that Shiny has access to special objects called input and output.

Using these objects in the special way defined by Shiny allows the server code to respond or **"react"** to changes in inputs.

This is reactive programming.

https://mastering-shiny.org/basic-reactivity.html

What have we learned so far?

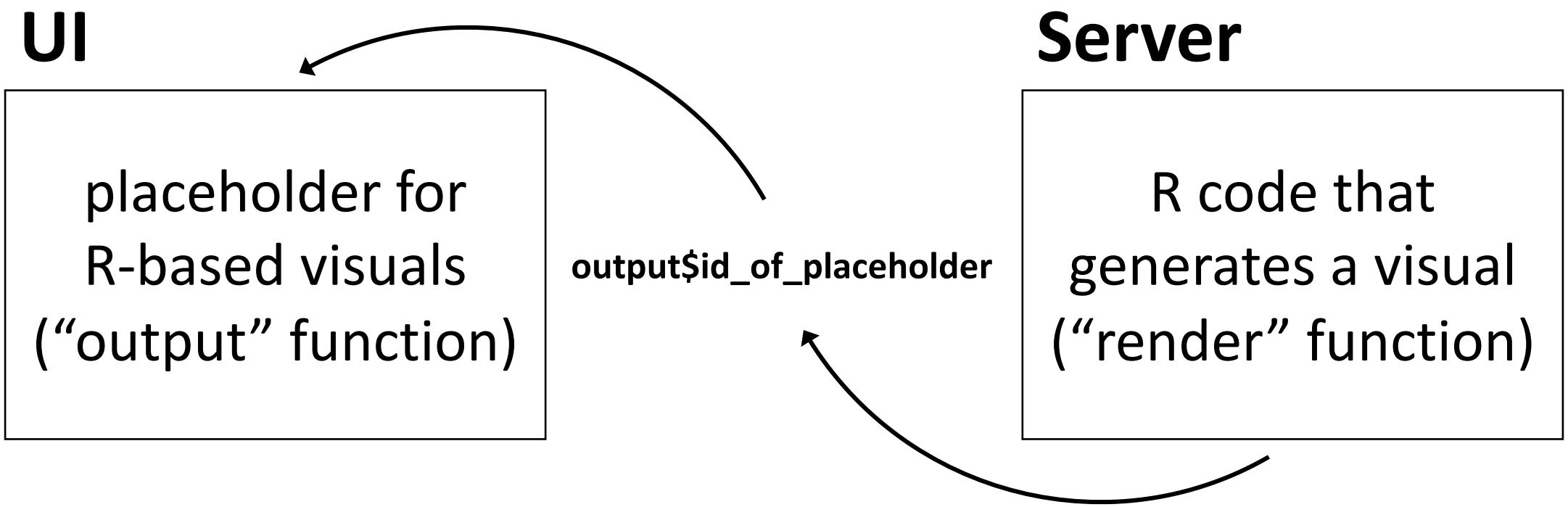
User Interface (UI)

the website people will see and interact with, including static content, inputs and (placeholders for) outputs

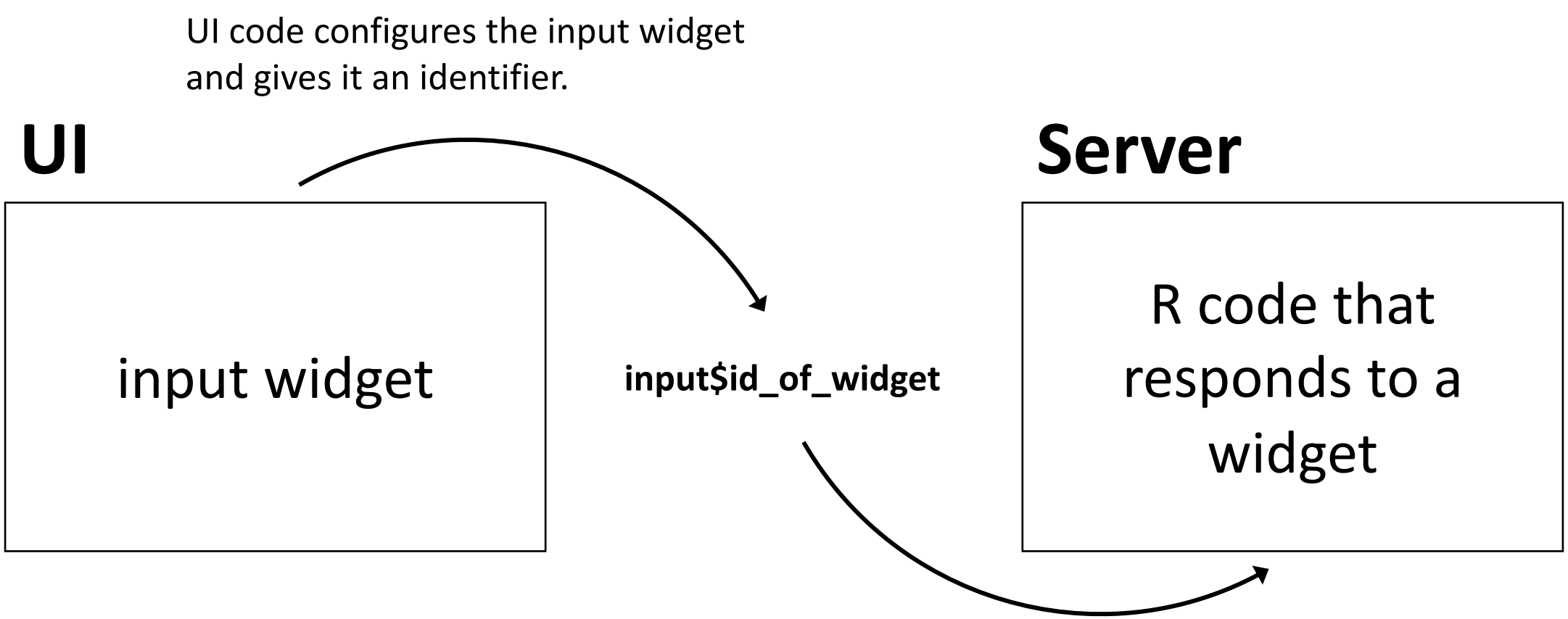
Server

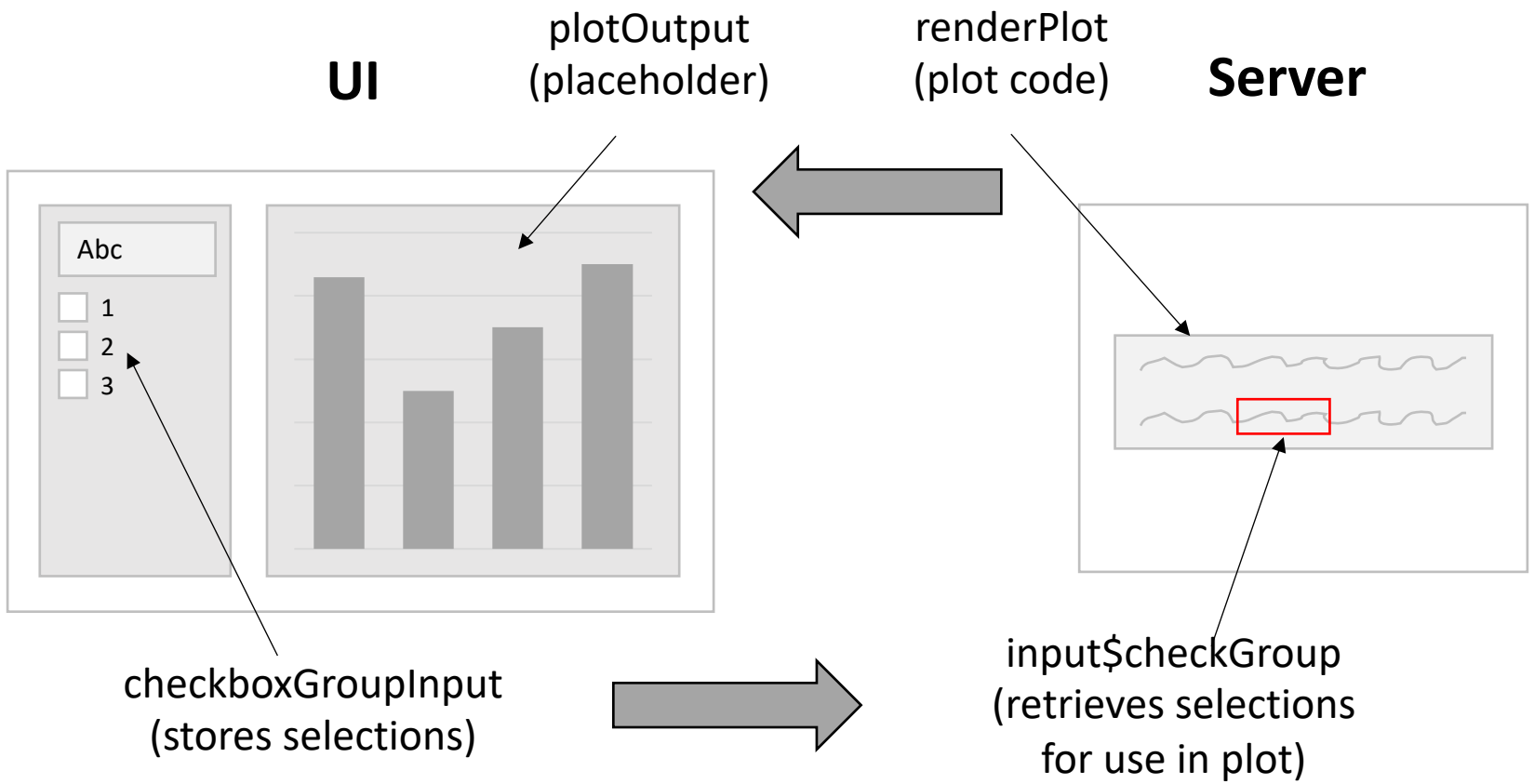
takes values from the inputs, does some calculations, and fills in the outputs

UI code specifies a placeholder to indicate where the R output should go and give the output an identifier.



Server code saves R output to an object called "output", using the same ID listed in the placeholder.


UI code configures the input widget
and gives it an identifier.
UI
Server
input widget
input$id_of_widget
R code that
responds to a
widget
Server code watches the object called
“input”, using the same ID listed in the UI.
R code is rerun when the input changes.


UI
plotOutput
(placeholder)
renderPlot
(plot code)
Server
Abc
1
2
3
checkboxGroupInput
(stores selections)
input$checkGroup
(retrieves selections
for use in plot)

Exercise 6: App from scratch

- In RStudio, use the File menu to create a new file
- Select “Shiny Web App…”
- Use any name you like
- Leave “Single file (app.R)” selected
- Select any directory you like
- Click “Create”
- Explore the app structure

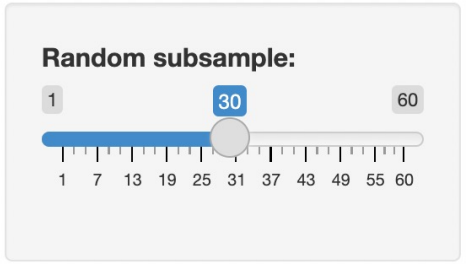
Discussion: What would you like to do with Shiny?

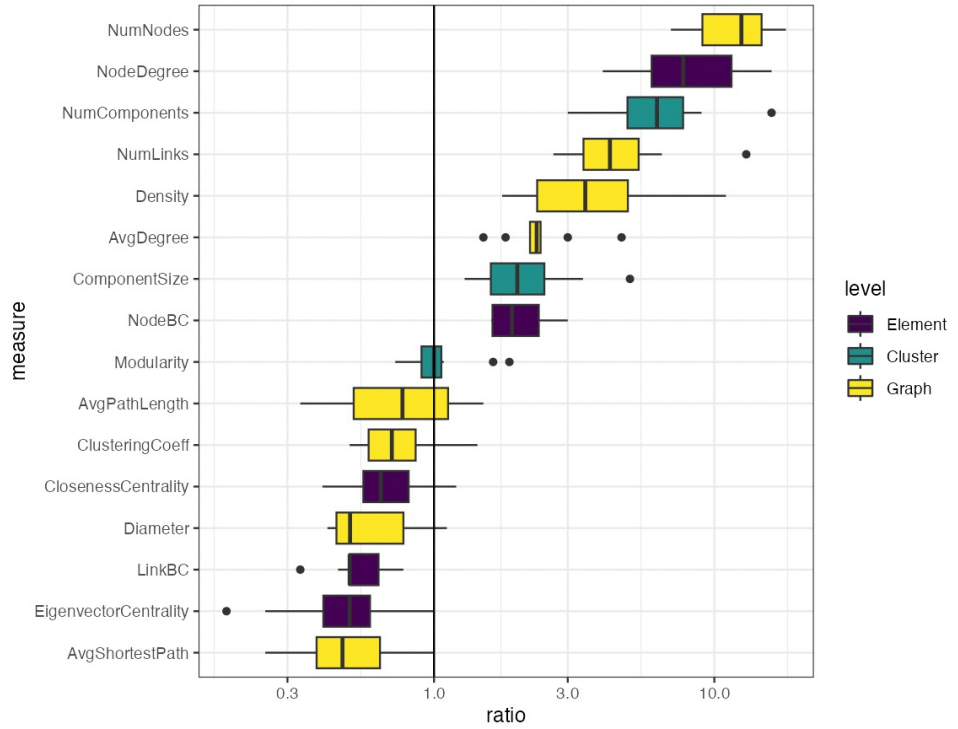
Other examples (in “examples” folder)

May require additional packages

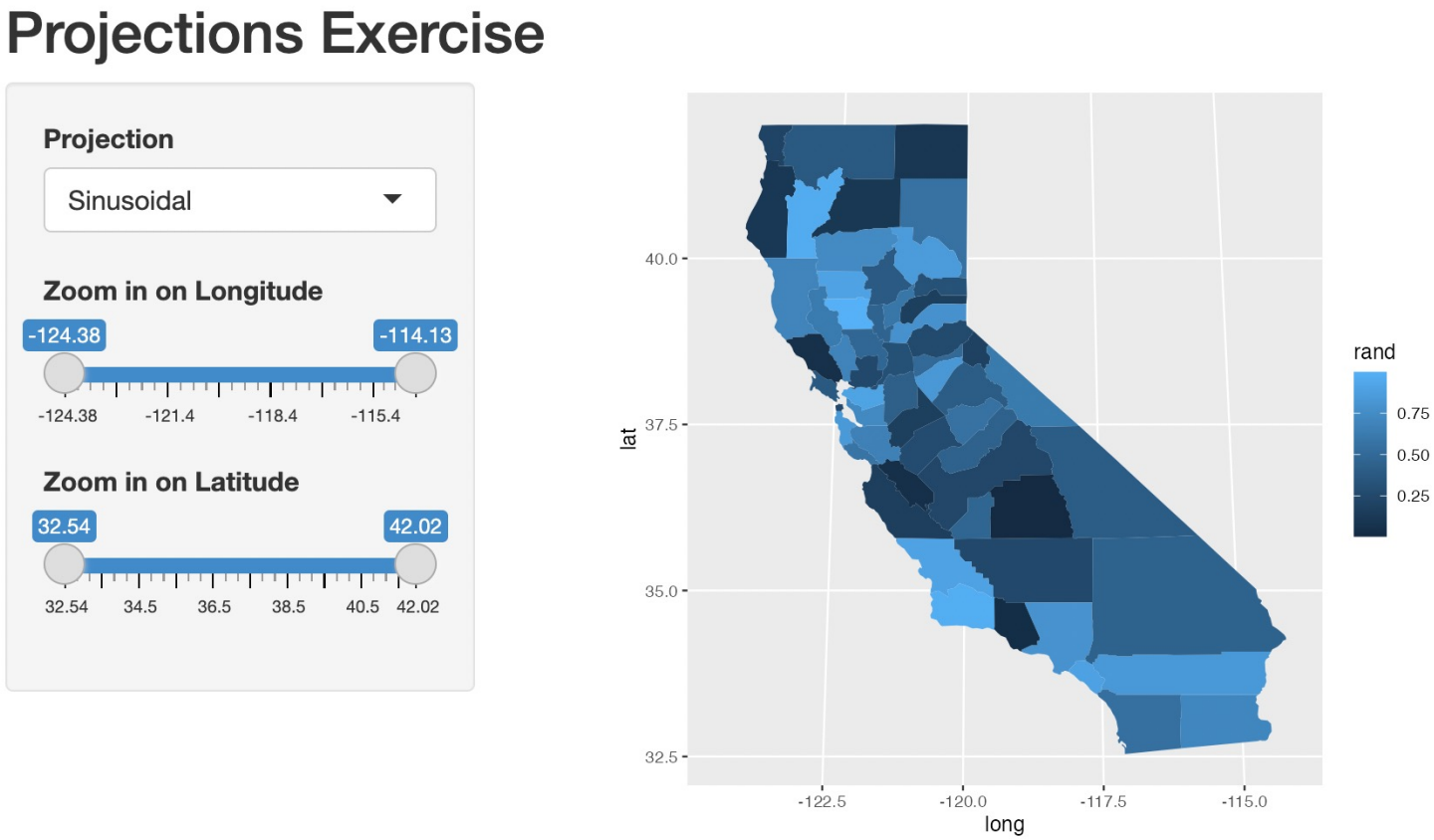
Ratings

Bootstrapping





Mapping


Projections Exercise
Projection
Sinusoidal
Zoom in on Longitude
-124.38
-114.13
-124.38
-121.4
-118.4
-115.4
Zoom in on Latitude
32.54
42.02
32.54
34.5
36.5
38.5
40.5
42.02
40.0
37.5
35.0
32.5
lat
-122.5
-120.0
-117.5
-115.0
long
rand
0.75
0.50
0.25

Graph Builder

ggplot Chart Builder

Chart design

Choose a variable for the X-axis:

mpg

Choose a variable for the Y-axis:

disp

Choose a built-in theme:

theme_grey

Annotation

To place a text annotation on the graph, type the annotation below and click inside the chart to indicate the correct location.

Type an annotation:

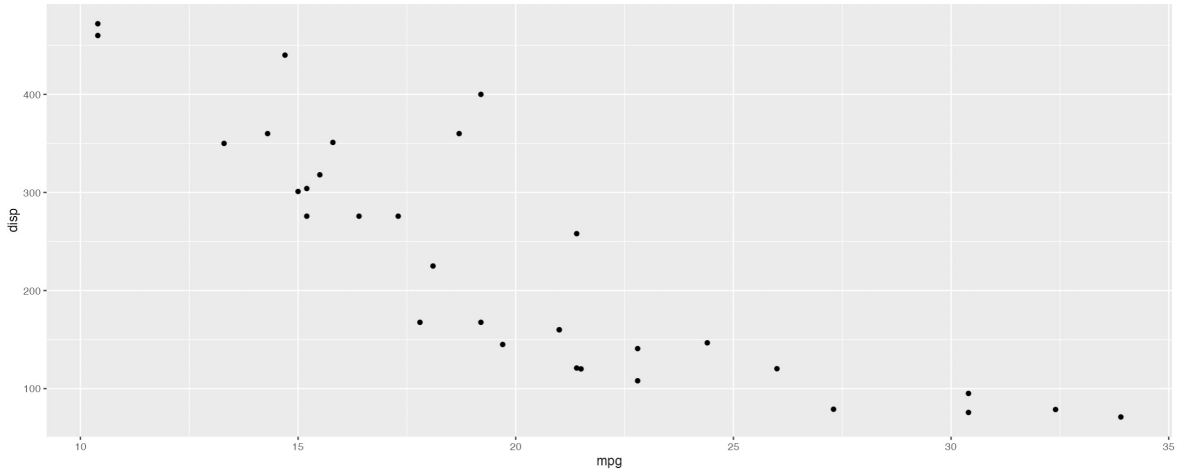
Change the size of the text:

1 8 20

1 3 5 7 9 11 13 15 17 19 20

Interactivity

☐ Make it interactive?



Code to create this plot:

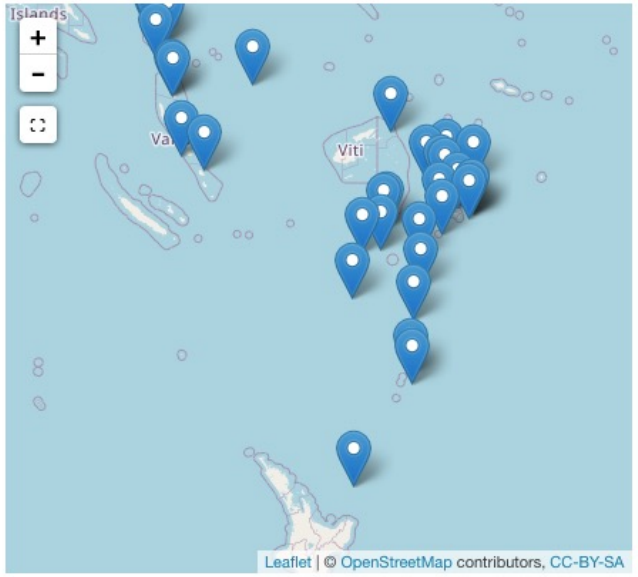
```
ggplot(mtcars, aes(x=mpg, y=disp)) + geom_point() + theme_grey()
```

Afternoon break

Coordinated Views

Views that share data

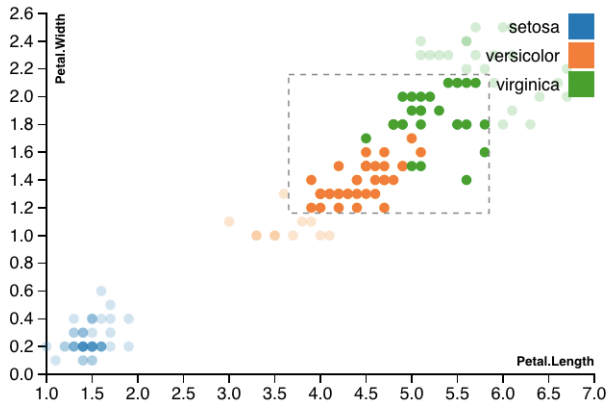
- Each view should be relatively simple, have a specific purpose
- Views can work together to explore complex interactions
- How can Shiny help us display multiple views of the same data?

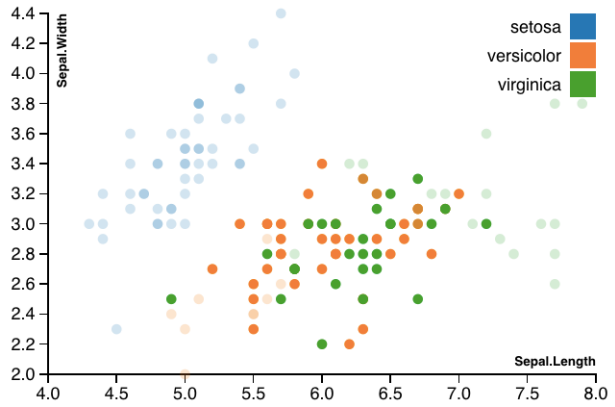


Search:

	lat	long	depth	mag	stations
308	-22	180.53	583	4.9	20
873	-11.02	167.01	62	4.9	36
277	-23.33	180.18	528	5	59
752	-21.29	185.77	57	5.3	69
352	-12.01	166.29	59	4.9	27
354	-30.17	182.02	56	5.5	68
168	-19.89	183.84	244	5.3	73
474	-10.79	166.06	142	5	40
338	-27.19	182.18	69	5.4	68

Showing 1 to 10 of 32 entries (filtered from 100 total entries)





Option 1: Duplication

For each output that needs to be reactive to the input, repeat the same calculation.

```
output$a <- renderPlot({
  ...
  data %>%
    filter(column == input$value)
  ...
  })

output$b <- renderTable({
  ...
  data %>%
    filter(column == input$value)
  ...
  })
```

Option 2: reactive()

- Unbeknownst to us, we've been working in a "reactive context" every time we use a "render" function
- If we want to do some calculations in R that involve reactivity, but we don't necessarily want to create a visual output right away, we can use reactive() to store something that reacts to inputs
- **When you use something created by reactive(), you need to add () to the name**

```
variable <- reactive({
  ...
  data %>%
    filter(column == input$value)
  ...
  })

output$a <- renderPlot({
  ...
  variable()
  ...
  })

output$b <- renderTable({
  ...
  variable()
  ...
  })
```

Exercise 7: Reactive Data

- Create an app with a sidebar layout
- Add a select input with all of the values of the mtcars “cyl” column
- Add a plot placeholder and a table (or data table) placeholder
- In the server, create a function that filters the mtcars dataset based on the value selected in the input
- Use this same dataset to render both the plot and table outputs

Using charts as inputs

Turning a plotOutput into an input

The plotOutput function has a few additional (powerful!) options.

- **click**: when someone clicks on the plot, store the x & y coordinates in input under the id specified
- **dblclick**: when someone double clicks, story the x & y coordinates
- **hover**: when someone hovers, store the x & y coordinates
- **brush**: when someone clicks and drags to draw a box on the plot, store the x and y coordinate ranges

Additional related functions

When you add a click or brush to a plotOutput, you can use the following functions in the server code:

- **nearPoints()**: returns rows from a data frame which are near a click, hover, or double-click
- **brushedPoints()**: returns rows from a data frame which are under a brush

Exercise 8: Plots and brushes

- Create an app with two plots
- Add a brush to one plot
- In the other plot, tie the brush to a layer that adds red points

How to learn more

Advanced features and concepts

- reactiveVal() and reactiveValues(): variables that are created in the server but can trigger code to re-run when updated
- observe() and observeEvent(): server code that reacts to changes in inputs but doesn't get saved as a variable
- modules: for complicated applications, you might want to break up your code into smaller chunks called modules
- debugging: when you have trouble figuring out why your app isn't working, you might want to use some debugging options like browser() and breakpoints

Shiny resources

- Mastering Shiny book
- Shiny Tutorial
- Shiny Articles
- Shiny function reference
- Shinyapps.io
- rstudio::conf 2022 workshop: Getting Started with Shiny
- Shiny in Production (slides), Shiny in Production (book)
- Big Book of R: Shiny

Thanks for your time today!

angela.zoss@duke.edu