Visualization for Data Science in R

Angela Zoss

Data Matters 2021

https://www.angelazoss.com/RVis-2Day/

Objectives/Outline

Day 1: Static visualizations

- Visualization and data science
- Basic ggplot2 syntax
- Basics of geoms and aes
- Manipulating data
- Categorical variables
- Advanced topics: mapping, saving charts out

Day 2: Interactivity

- Day 1 Review, sample projects
- Simple interactive plots
- Arranging charts into dashboards
- Incorporating Shiny elements into documents, dashboards
- Advanced topics: full Shiny apps

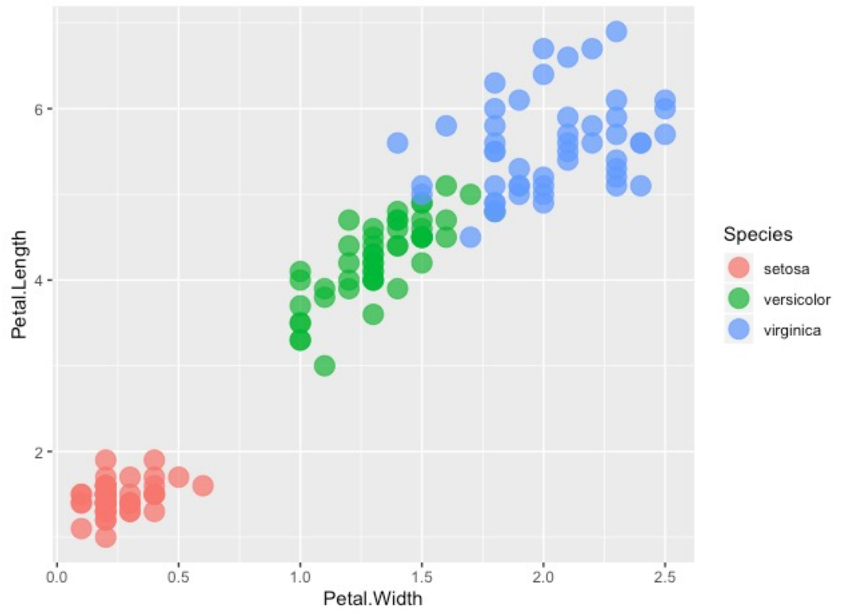
Day 1 Review

Example plot

"iris"

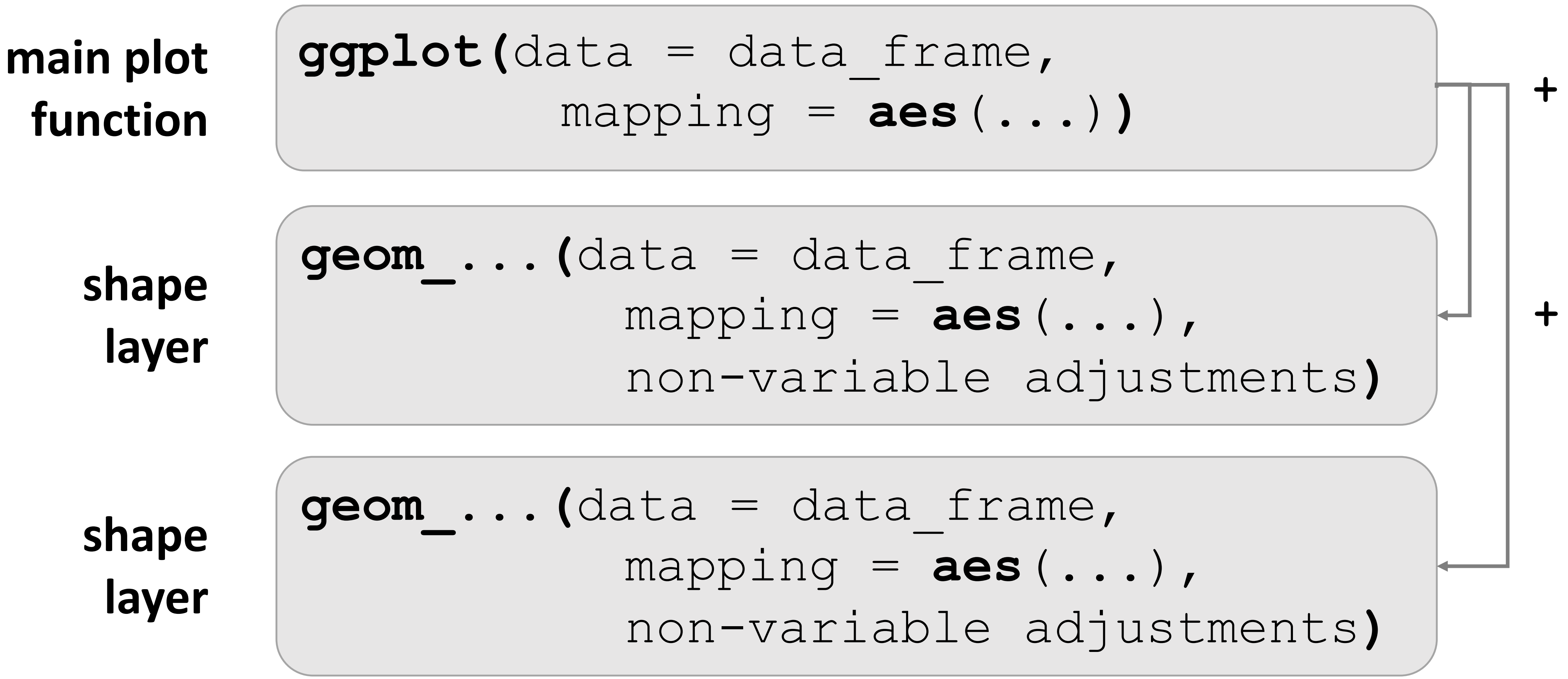
Petal.Width	Petal.Length	Species
0.3	1.4	setosa
1.3	4.0	versicolor
2.1	5.7	virginica

```
ggplot(data=iris) +
  geom_point(
     mapping=aes(x=Petal.Width,
                 y=Petal.Length,
                 color=Species),
     size=5, alpha=.75)
```



General pattern

data and aesthetics will carry through from main function to shape layers



main plot function

```
ggplot(data = data_frame,
       mapping = aes(...))
```

shape layer

```
geom_...(data = data_frame,
         mapping = aes(...),
         non-variable adjustments)
```

shape layer

```
geom_...(data = data_frame,
         mapping = aes(...),
         non-variable adjustments)
```

geom vs. scale vs. theme

Adding something that will appear
inside the **chart coordinate space**?

You will (almost always) be adding a **geom**!

Changing the way a **variable is displayed**?
(e.g., different axis breaks, different color mapping)

You will be adding a **scale**!

Changing the **look and feel** of the chart?

You will be adding or making changes to a **theme!**

Sample Projects

Designing tools for data exploration

Supporting data exploration

Output

Picking the right visual elements

Input

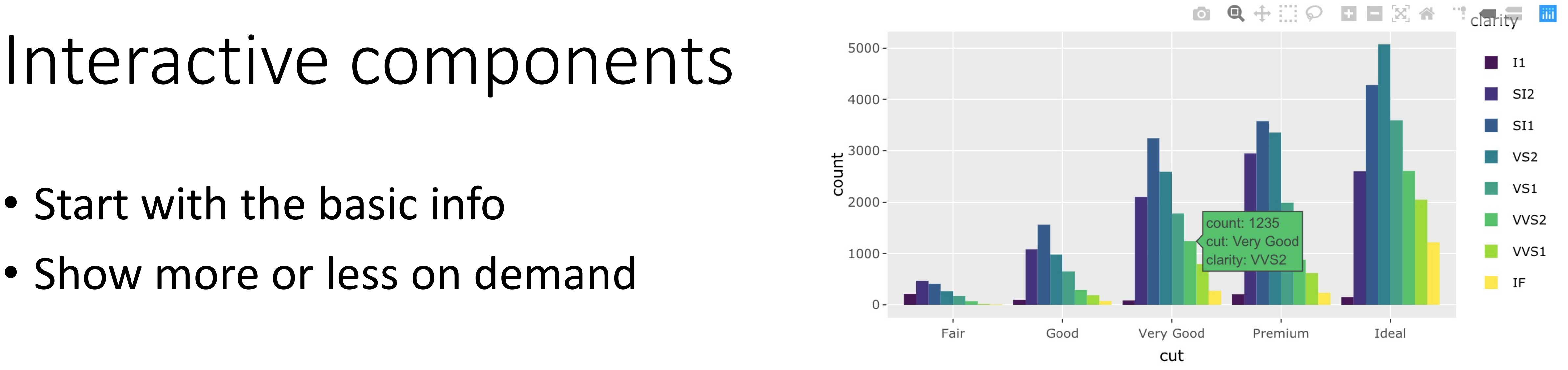
Giving users the right controls

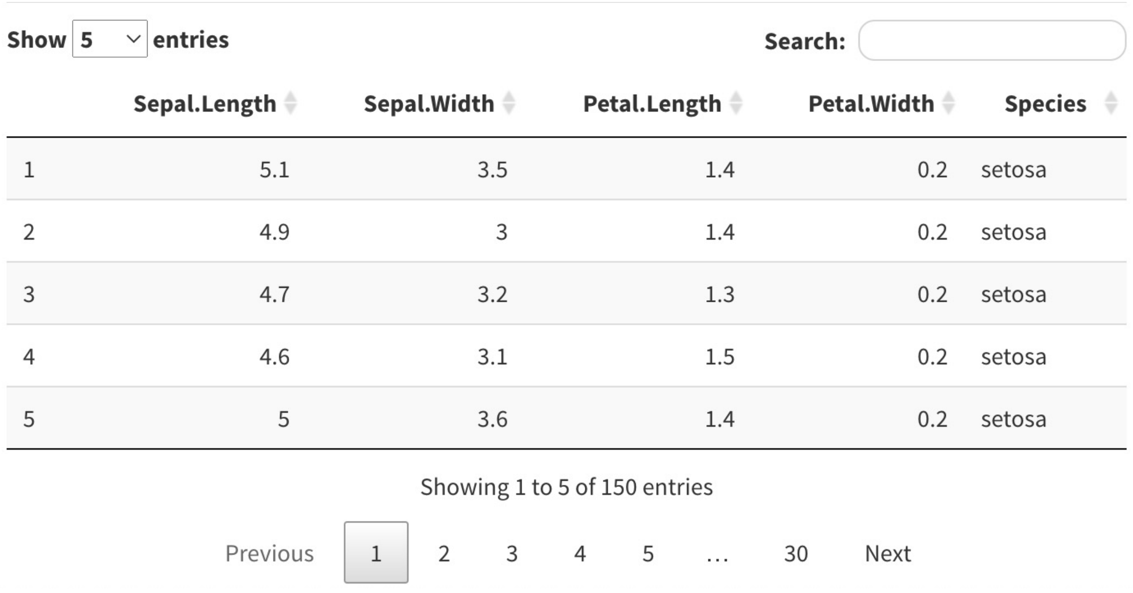
Layout

Arranging everything in the right place

Interactive components

- Start with the basic info
- Show more or less on demand



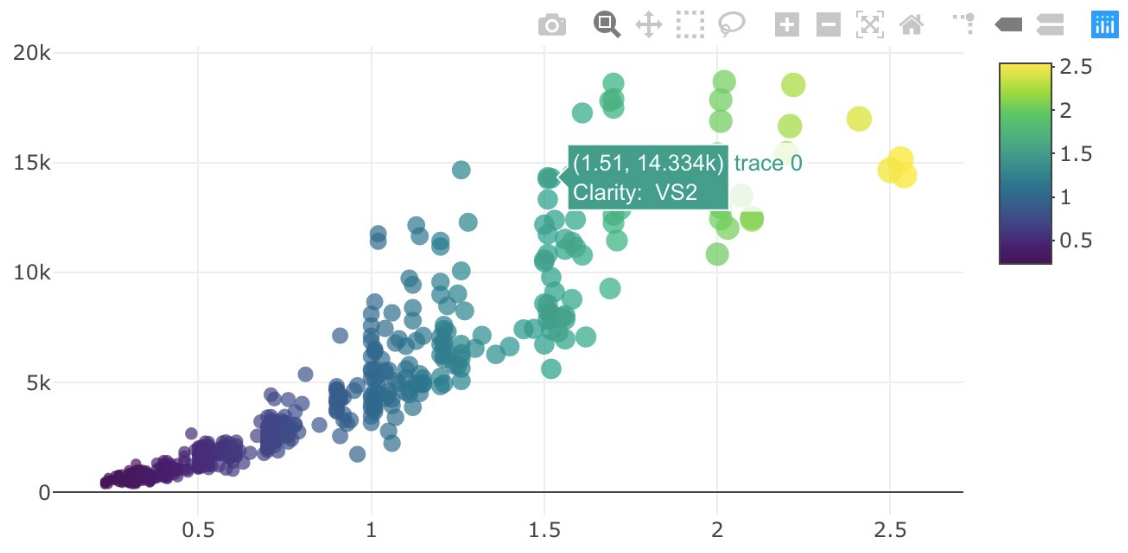


Show 5 entries Search:

	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

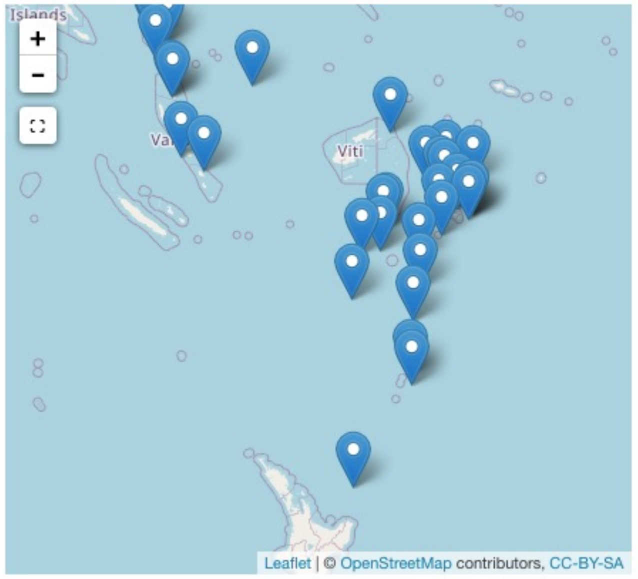


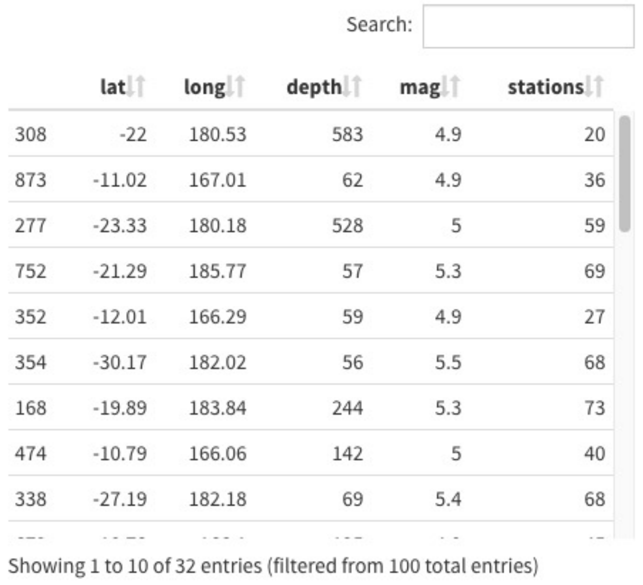
https://www.htmlwidgets.org/

http://gallery.htmlwidgets.org/

Coordinated views

- Different parts of the story, working together

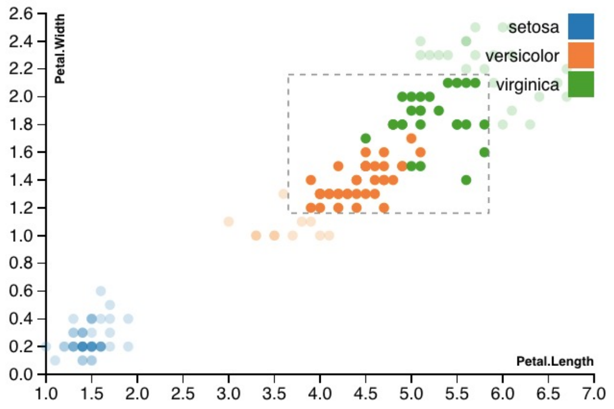


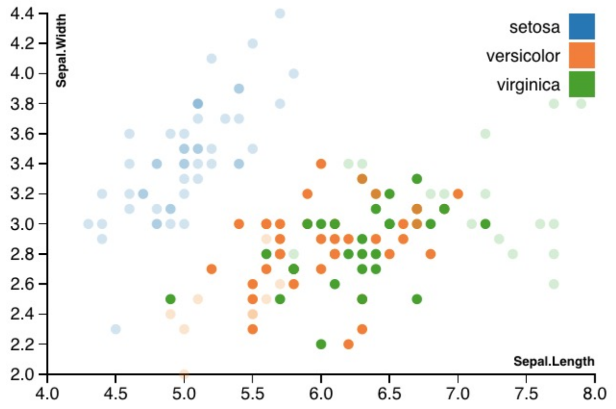


Search:

	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)

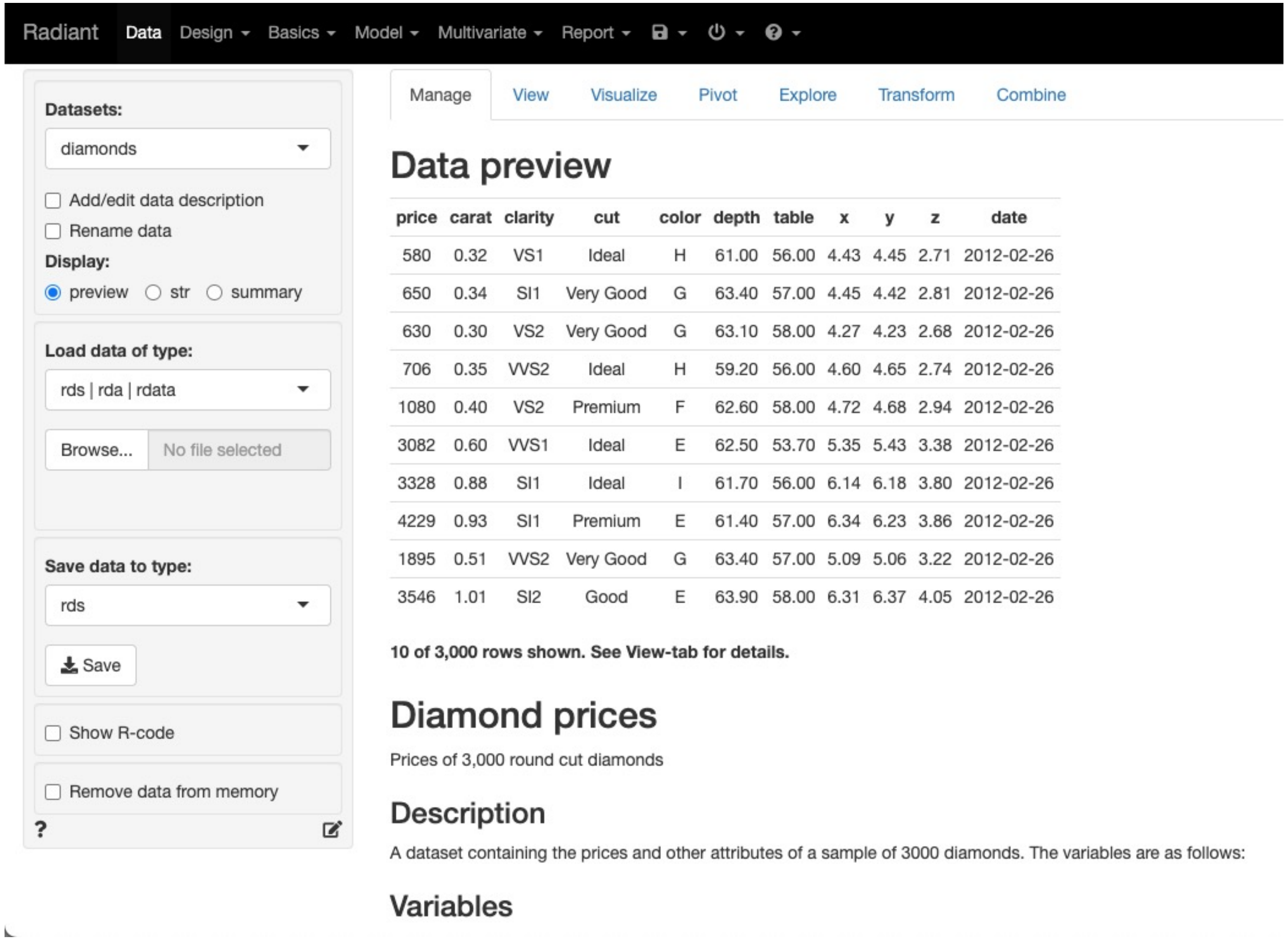




https://rstudio.github.io/crosstalk/

Responding to user input

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



Datasets: diamonds

[ ] Add/edit data description

[ ] Rename data

Display: [✓ preview] str summary

Load data of type: rds | rda | rdata

No file selected

Save data to type: rds

[ ] Show R-code

[ ] Remove data from memory

Data preview

price	carat	clarity	cut	color	depth	table	x	y	z	date
580	0.32	VS1	Ideal	H	61.00	56.00	4.43	4.45	2.71	2012-02-26
650	0.34	SI1	Very Good	G	63.40	57.00	4.45	4.42	2.81	2012-02-26
630	0.30	VS2	Very Good	G	63.10	58.00	4.27	4.23	2.68	2012-02-26
706	0.35	VVS2	Ideal	H	59.20	56.00	4.60	4.65	2.74	2012-02-26
1080	0.40	VS2	Premium	F	62.60	58.00	4.72	4.68	2.94	2012-02-26
3082	0.60	VVS1	Ideal	E	62.50	53.70	5.35	5.43	3.38	2012-02-26
3328	0.88	SI1	Ideal	I	61.70	56.00	6.14	6.18	3.80	2012-02-26
4229	0.93	SI1	Premium	E	61.40	57.00	6.34	6.23	3.86	2012-02-26
1895	0.51	VVS2	Very Good	G	63.40	57.00	5.09	5.06	3.22	2012-02-26
3546	1.01	SI2	Good	E	63.90	58.00	6.31	6.37	4.05	2012-02-26

10 of 3,000 rows shown. See View-tab for details.

Diamond prices

Prices of 3,000 round cut diamonds

Description

A dataset containing the prices and other attributes of a sample of 3000 diamonds. The variables are as follows:

Variables

https://shiny.rstudio.com/

https://shiny.rstudio.com/gallery/radiant.html

Interactive components

Why make charts interactive?

- Easier for data exploration
  - Drill-down to data subsets of interest
  - Details on demand
  - Customize look-and-feel of chart
- Can be more engaging for users

Visual information seeking mantra

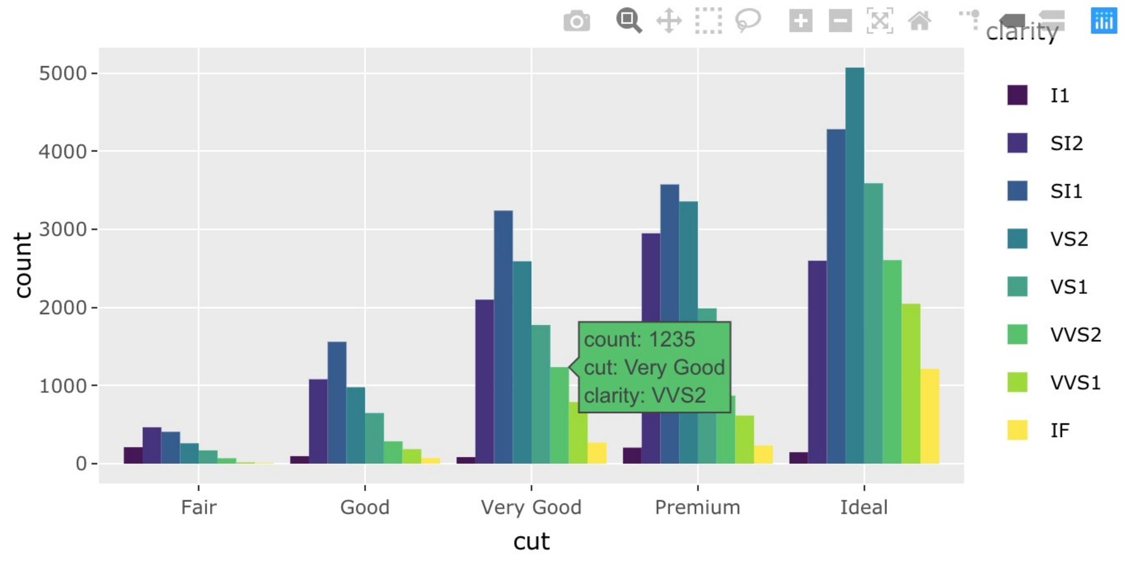
Overview first,
zoom and filter,
then details-on-demand

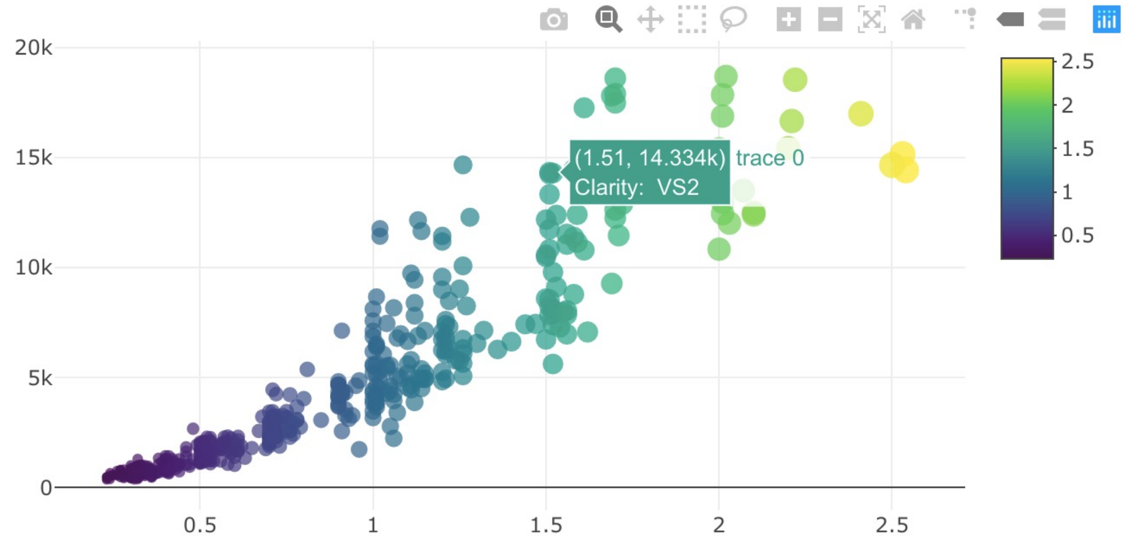
Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualization.
In *VL '96 Proceedings of the 1996 IEEE Symposium on Visual Languages.*

Interactivity in R Markdown

- R Markdown gets compiled into HTML
- Some R packages can create interactive elements by converting R output to HTML/JavaScript code in the final document
- We will use the **plotly** package to create interactive charts

http://www.htmlwidgets.org/





Other interactive chart packages

- ggiraph for extending ggplot2 with interactive geoms
- rCharts for an R version of Polycharts, NVD3, and MorrisJS
- rBokeh for an R version of Bokeh
- altair for an R version of Altair
- leaflet for interactive maps

Exercise 1: Make yesterday’s charts interactive

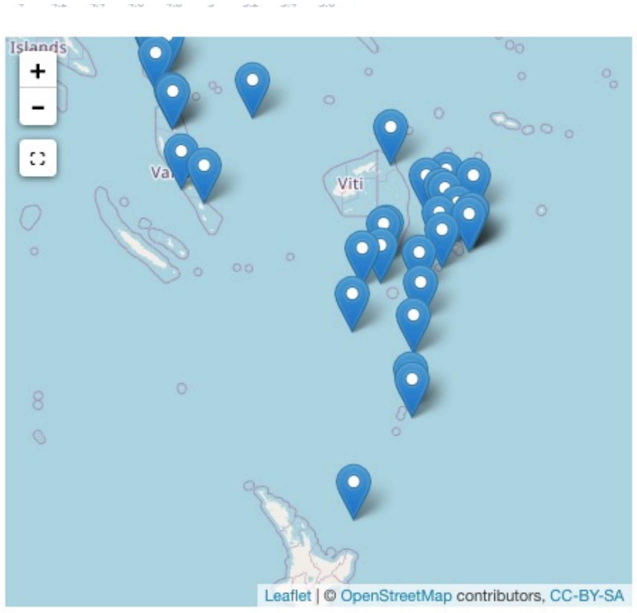
Exercise 2:
DT for interactive data tables

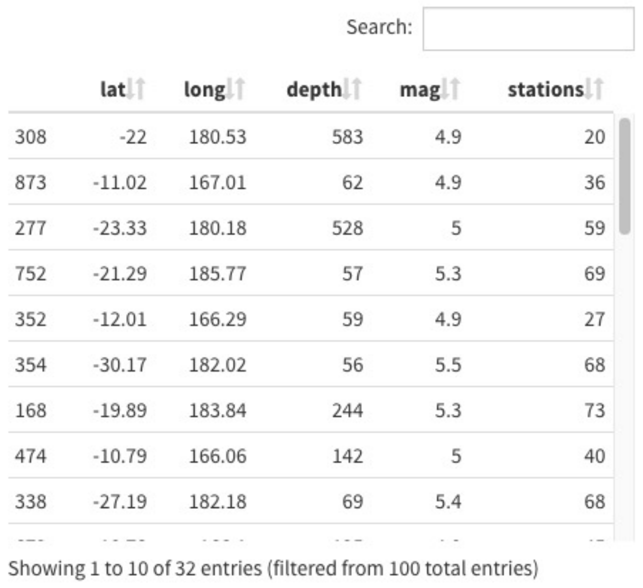
Coordinated Views

Views that share data

- Each view should be relatively simple, have a specific purpose
- Views can work together to explore complex interactions
- The **Crosstalk** package connects interactive components together

https://rstudio.github.io/crosstalk/

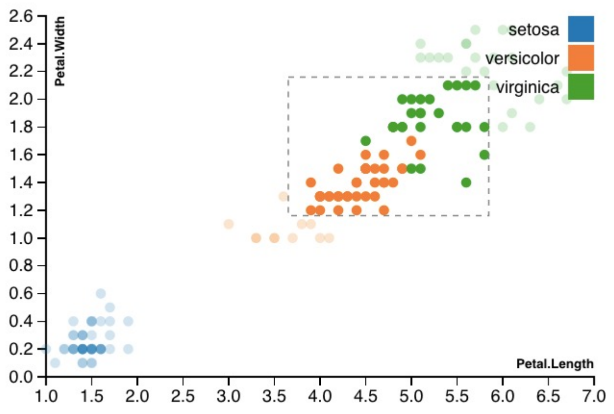


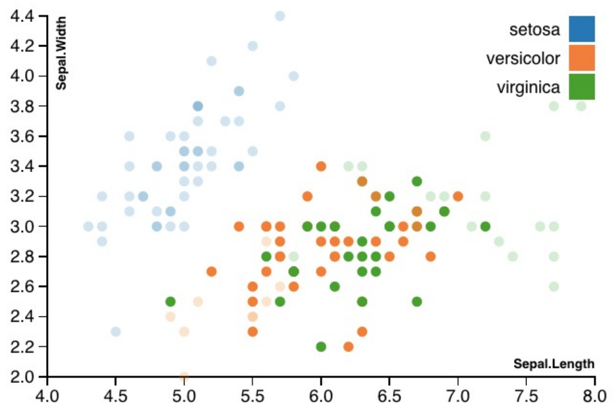


Search:

	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)



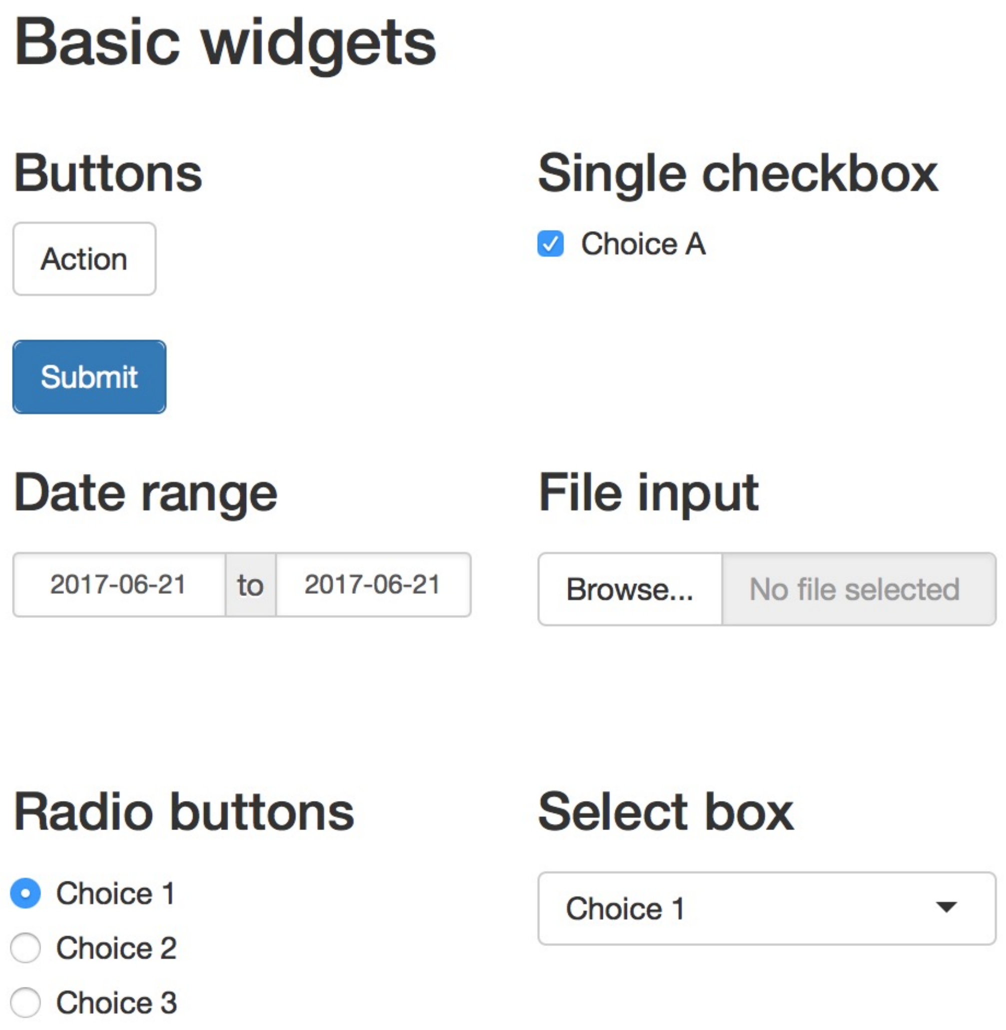


Exercise 3: Combining interactive components with Crosstalk

Responding to user input

Input controls to guide exploration

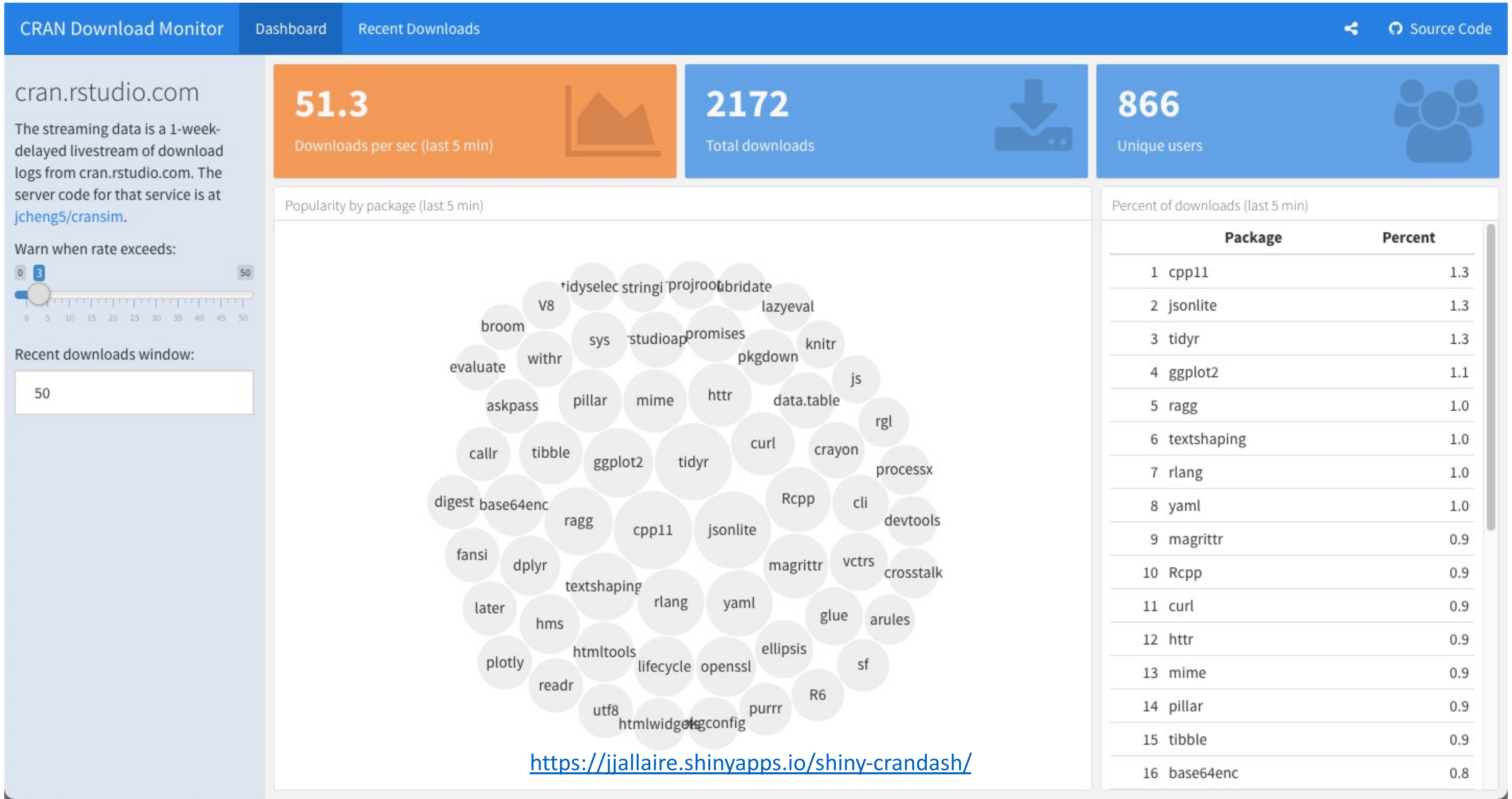
- For more complex data exploration, you may need input from the user
- Input controls can gather different kinds of information from the user, from free text to buttons to date ranges
- Simple input processing can happen within a standalone website, but for complex data processing, the input may need to feed back into a real R calculation (Shiny)



https://rstudio.github.io/crosstalk/ https://shiny.rstudio.com/

Dashboards in R Markdown

What is a dashboard?



cran.rstudio.com

The streaming data is a 1-week-delayed livestream of download logs from cran.rstudio.com. The server code for that service is at jcheng5/cransim.

Warn when rate exceeds: 3

Recent downloads window: 50

51.3 Downloads per sec (last 5 min)

2172 Total downloads

866 Unique users

Popularity by package (last 5 min)

Percent of downloads (last 5 min)

	Package	Percent
1	cpp11	1.3
2	jsonlite	1.3
3	tidyr	1.3
4	ggplot2	1.1
5	ragg	1.0
6	textshaping	1.0
7	rlang	1.0
8	yaml	1.0
9	magrittr	0.9
10	Rcpp	0.9
11	curl	0.9
12	httr	0.9
13	mime	0.9
14	pillar	0.9
15	tibble	0.9
16	base64enc	0.8

https://jjallaire.shinyapps.io/shiny-crandash/

“Normal” R Markdown

- R Markdown elements like headings, text

```
# Heading 1
## Heading 2
Regular text
* Bulleted text
```

Note: Comments work like HTML
<!--HTML Comment style -->

- Code chunks

~~~
```{r}

```
~~~

Markdown for flexdashboards

````
Page
===================

Column (or Row)
--------------------------

### Chart titles
````

````
Regular text
* Bulleted text
````

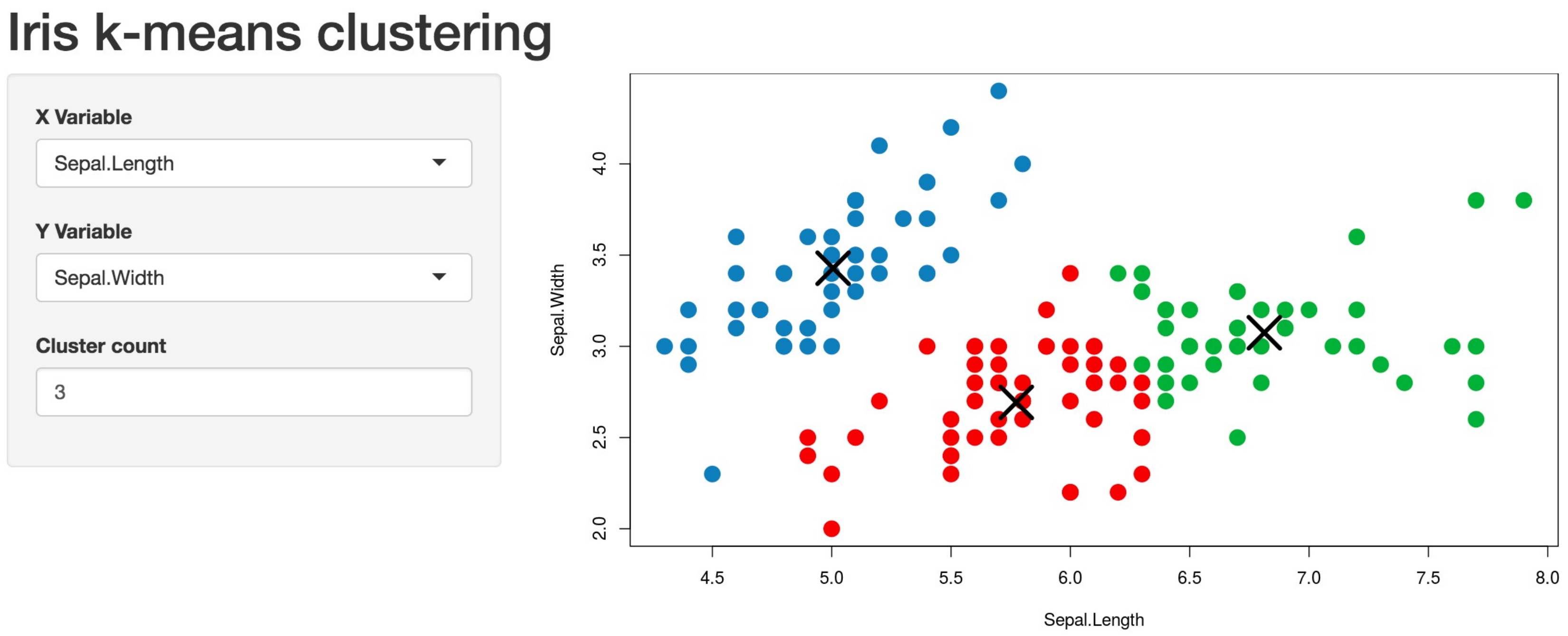
~~~
```{r}

```
~~~

Exercise 4: Arrange Crosstalk elements in flexdashboard

Shiny

What is Shiny?

An interactive interface onto an R program



Iris k-means clustering

X Variable

Sepal.Length

Y Variable

Sepal.Width

Cluster count

3

http://shiny.rstudio.com/

How can you use Shiny for visualization?

- Use Shiny to control some kind of simulation interactively, then visualize the results
- Use Shiny to change components within the chart (e.g., switch the mappings)
- Use Shiny to filter data to subsets to highlight patterns
- Change type of regression, plot results

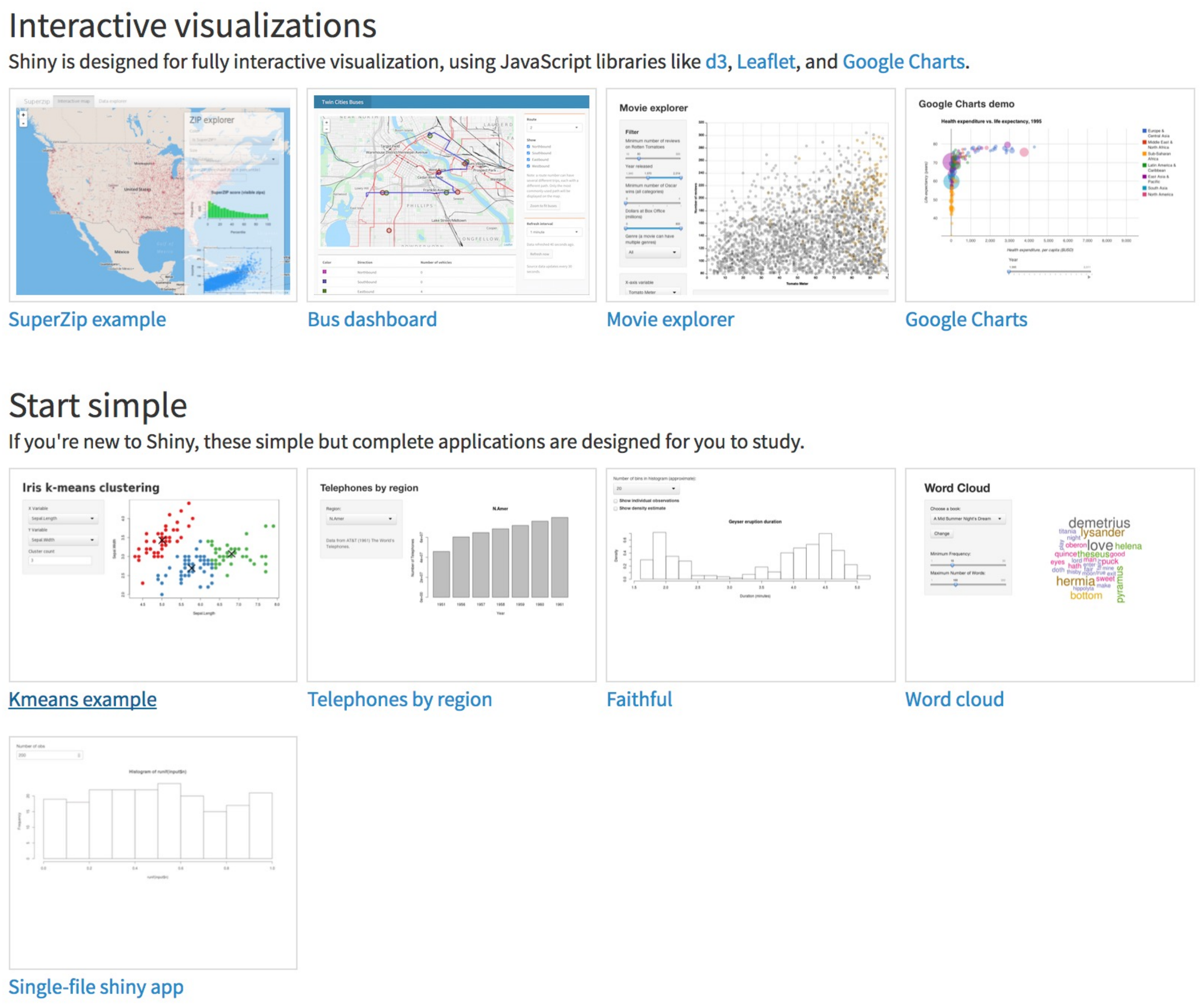
Crosstalk

- Small number of inputs
- Small number of outputs (htmlwidgets)
- Interactions between inputs and outputs are fairly simple
- Layout restricted to what can be written in R Markdown
- Compiles to a simple website which can be hosted anywhere

Shiny

- Large number of inputs
- Large number of outputs
- Can program very complex interactions between inputs and outputs
- Flexible layout based on standard web interface elements
- Shiny app has to be hosted on a special Shiny server

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.rstudio.com/gallery/

Adding shiny controls to flexdashboards

```
Page
==================

Column {.sidebar}
---------------------------

### Chart titles
```

```
Regular text
* Bulleted text
```

~~~
```{r}

(including Shiny input, render objects)

```
~~~

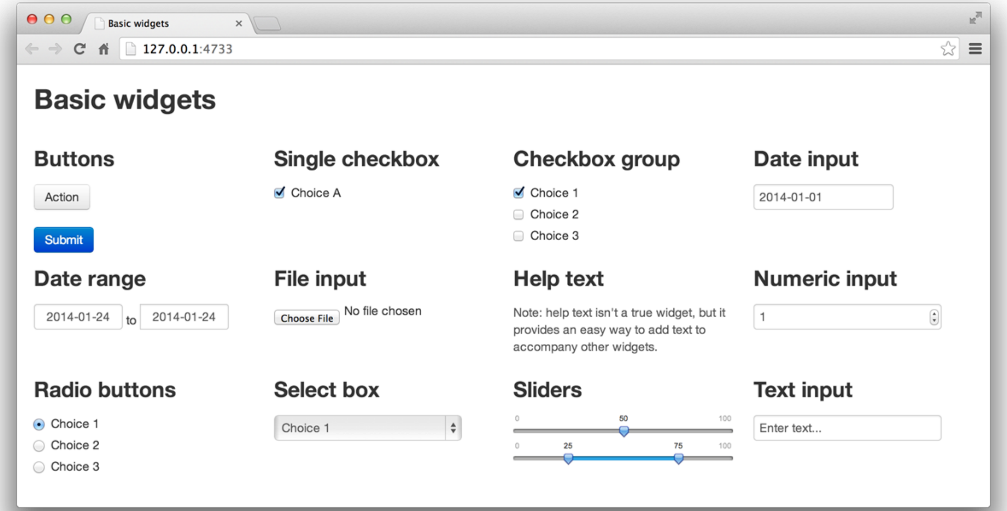
Components

Some kind of **input widget**

(e.g., selectInput, sliderInput)

Some kind of **render object**

(e.g., renderPlot, renderTable)



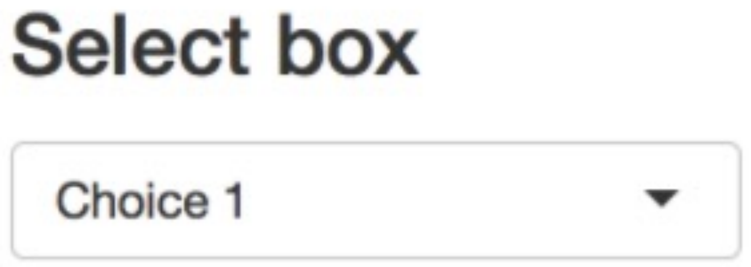
Note: help text isn't a true widget, but it provides an easy way to add text to accompany other widgets.

renderPlot wraps around something like a ggplot() plot

a plot can read data from the input widget using input$inputId

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 = "state",
            label = "Choose a state:",
            choices = c("NY", "NJ", "CT", "WA", "OR",
                        "CA", "MN", "WI", "IA")
```

Exercise 5:
Shiny inputs in dashboards

Shiny Apps

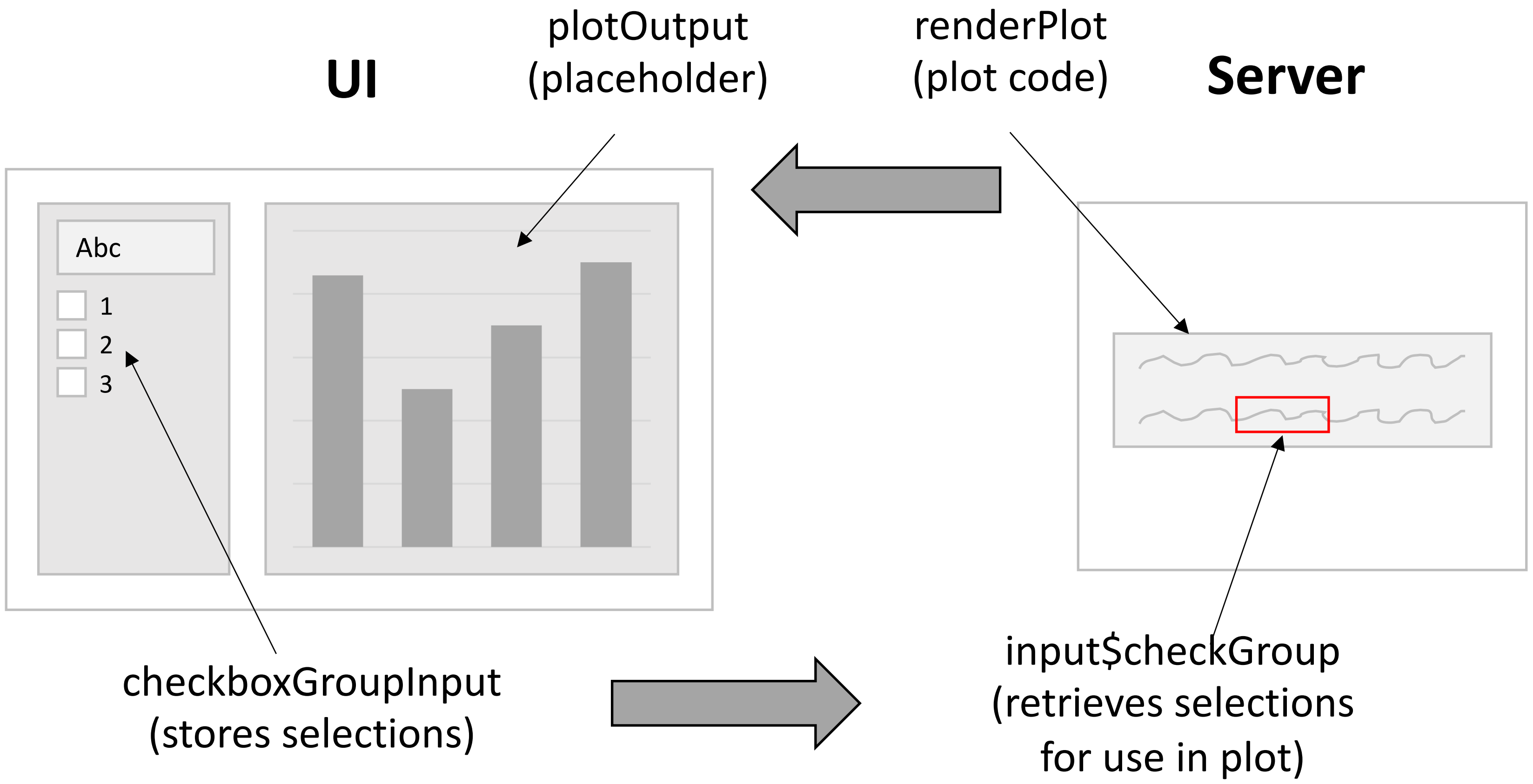
How do you build a Shiny app?

User Interface (UI)

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

Server

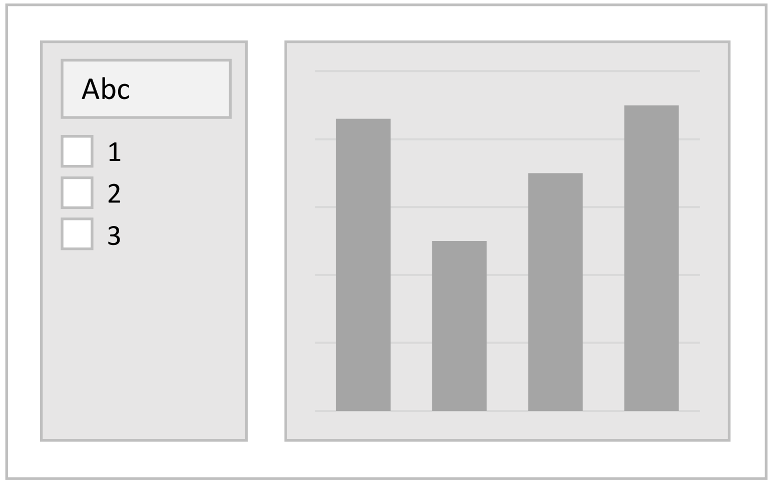
takes values from the inputs, does some calculations, and fills in the outputs


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

Step 1: Create the interface

What to put in the UI?

- Layout containers
- Input widgets
- Placeholders for reactive output
- Extra text/HTML elements

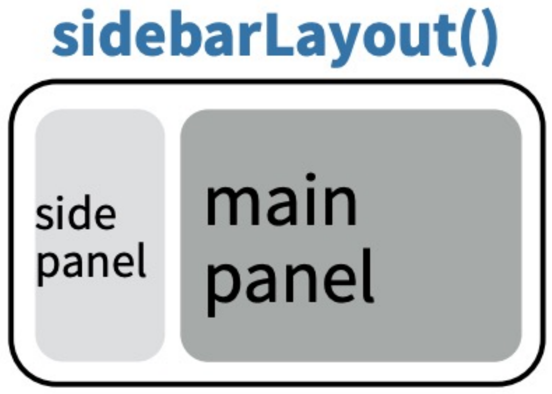


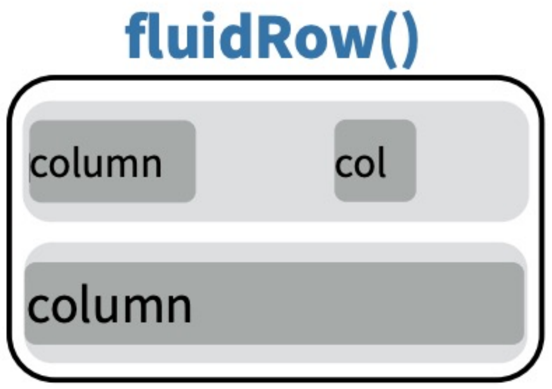
Page layout containers

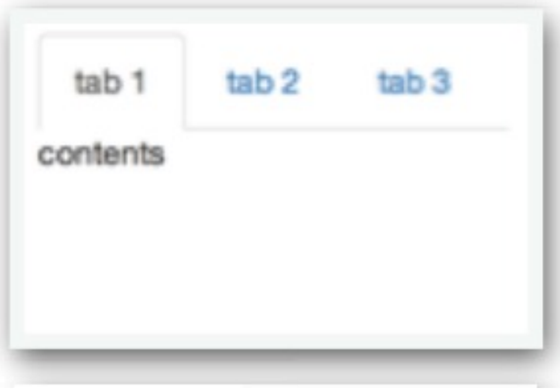
1. fluidPage
   - titlePanel
   - sidebarLayout
     - sidebarPanel
     - mainPanel
   - fluidRow
     - column
     - wellPanel
   - tabsetPanel
   - navlistPanel
2. fixedPage
   - fixedRow
3. navbarPage
   - tabPanel
   - navbarMenu
     - tabPanel

https://shiny.rstudio.com/articles/layout-guide.html

https://www.rstudio.com/resources/cheatsheets/ (Shiny)

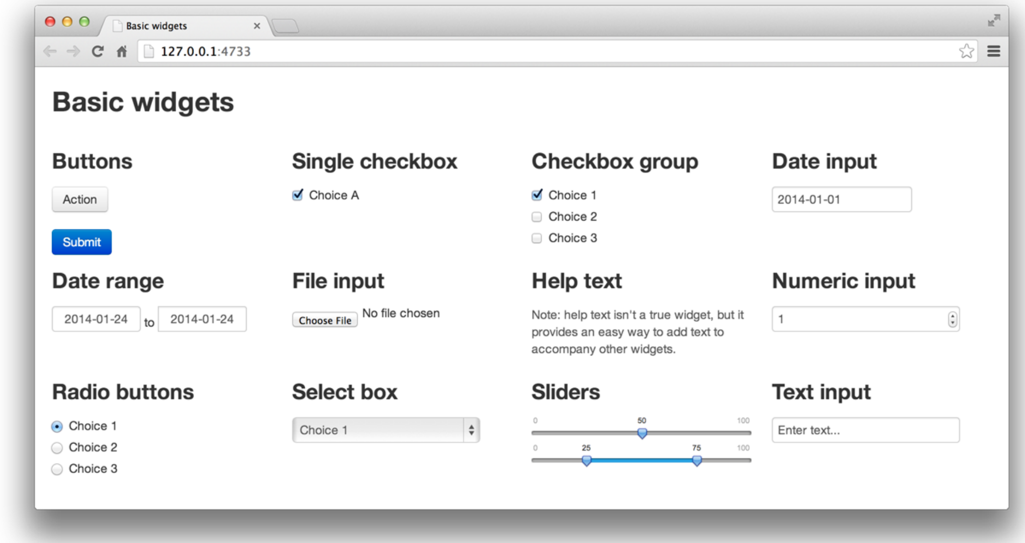






Input widgets

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



Note: help text isn't a true widget, but it provides an easy way to add text to accompany other widgets.

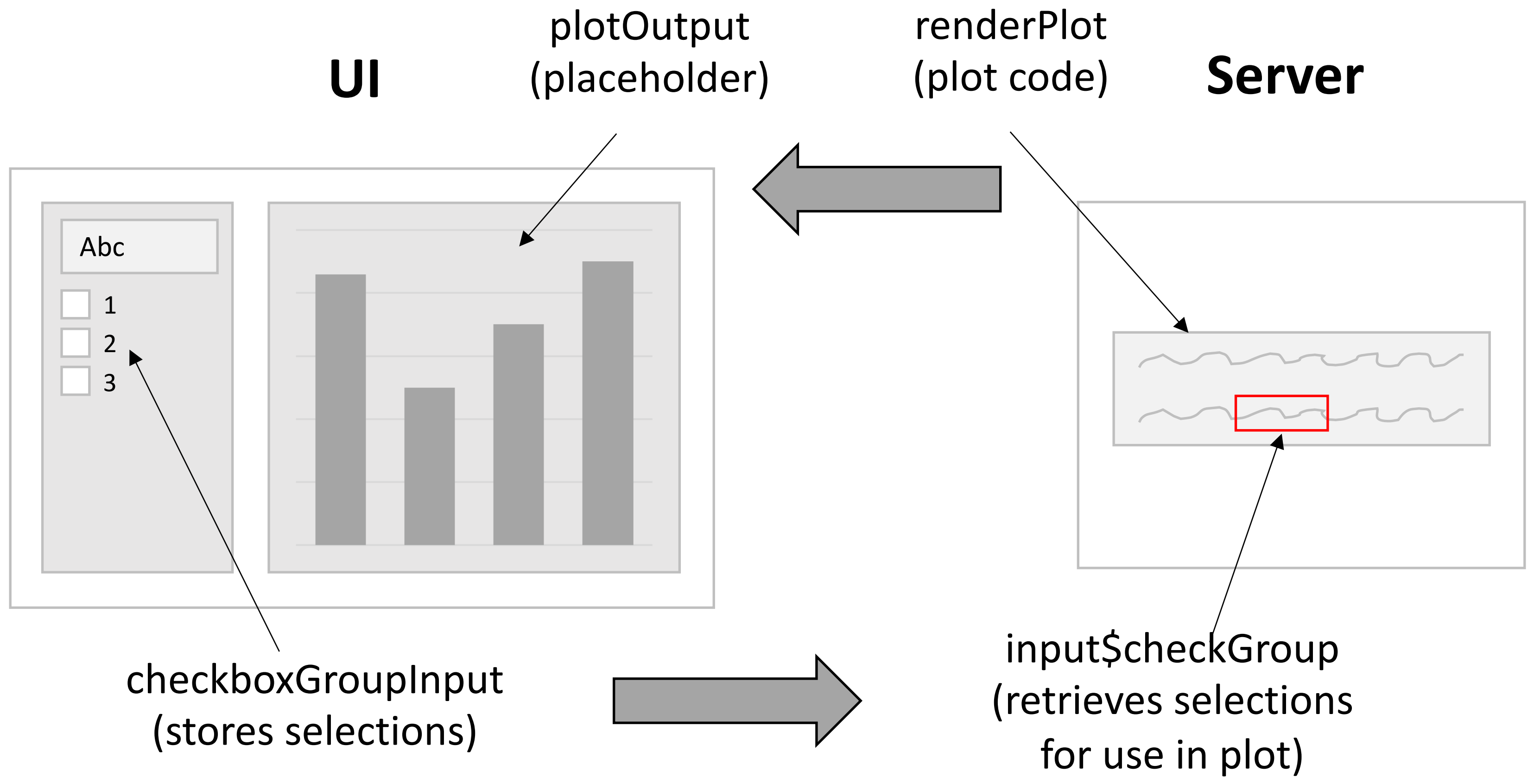
http://shiny.rstudio.com/tutorial/written-tutorial/lesson3/
http://shiny.rstudio.com/gallery/widget-gallery.html

Reactive output objects

UI	Server
htmlOutput	renderUI
imageOutput	renderImage
plotOutput	renderPlot
tableOutput	renderTable
textOutput	renderText
uiOutput	renderUI
verbatimTextOutput	renderPrint

http://shiny.rstudio.com/tutorial/written-tutorial/lesson4/

Step 2: Set up server to create dynamic objects


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

What to put in the server

- R code
- Render objects with same names and types as the ones listed in UI
- Input objects with the same names as the control widgets

UI:

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

textOutput("text1")
```

Server:

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

http://shiny.rstudio.com/tutorial/lesson4/

Step 3: Test

Running the app

Set options in RStudio:

- Window
- Viewer
- External

Exercise 6:
App from scratch

Create a new app

- File → New File → Shiny Web App…
- Set a name
- Use “Single File” application type
- Click “Run App” to see the default app
- Replace the code in ui and server to create your own app

Tips for building your first app

- Start with basic layout elements and static content, like plain text
- Add one output and connect it to something in the server (e.g., plotOutput/renderPlot), but don't use input$ in the plot yet
- Now create a control and add input$ to the server code
- You can save individual components as variables and then just use the variable names in your layout, if it gets confusing

Sharing an app

- Shiny Apps
  http://www.shinyapps.io/
- Shiny Server (free – host on your own server)
  https://github.com/rstudio/shiny-server/blob/master/README.md
- Shiny Server Pro (fee)
  https://www.rstudio.com/products/shiny/shiny-server/

Shiny resources

- Shiny Gallery
- Shiny Tutorial
- Shiny Articles
- Shiny function reference
- Shinyapps.io
- RStudio::conf 2019 workshop: Introduction to Shiny and R Markdown
- Shiny in Production (slides), Shiny in Production (book)
- Interactive web-based data visualization with R, plotly, and shiny
- Accessing and responding to plotly events in shiny

Thanks for your time this week!

angela.zoss@duke.edu