Shiny Workshop

Lingge Li

12/2/2016

What is Shiny

- A web framework for building applications that run in R
- Great for interactive data visualization
- Apps for education, data visualization and analysis

Server and UI

- ui.R defines everything you see

input widgets, plots, tables. . .

- server.R does the work

create plots, tables. . .

```
shinyServer(function(input, output) {

})

shinyUI(fluidPage(

))
```

Input

- Input is handled by specific widgets
- Each input has an id
- Can be accessed with input$id

`http://shiny.rstudio.com/gallery/widget-gallery.html`

Output

- Output is rendered in server as output$id
- Then displayed by id in ui

```r
shinyServer(function(input, output) {
  output$main_plot <- renderPlot({
    hist(faithful$eruptions)
  })
})

shinyUI(fluidPage(
  plotOutput(outputId = "main_plot")
))
```

Several types of output

- htmlOutput (uiOutput)
- plotOutput (imageOutput)
- tableOutput (dataTableOutput)
- textOutput (html)
- verbatimTextOutput (console)

Reactive environment

- Triggered when input changes
- Change output accordingly
- Rendering functions reactive

MoMA example

```
library(shiny)
library(ggplot2)

shinyServer(function(input, output) {
  Paintings <- read.csv("~/Paintings.csv")
  Paintings <- Paintings[Paintings$Height..cm.<500,]
  Paintings <- Paintings[Paintings$Width..cm.<1000,]
  Paintings <- Paintings[!is.na(Paintings$Height..cm.),]
  Paintings <- Paintings[!is.na(Paintings$Width..cm.),]

  output$painting <- renderPlot({
    ggplot(data=Paintings,
           aes(x=Width..cm., y=Height..cm.)) + geom_point()
      ggtitle('MoMA Painting Dimensions')
  })
})
```

Select color

- How can we use input$color in server

```r
shinyUI(fluidPage(
  mainPanel(plotOutput('painting')),
  selectInput('color', label = 'Point color',
              choices = list('black', 'red', 'blue'),
              selected = 'black')
))
```

Only Picasso

- Maybe extend this with selectInput

```
checkboxInput('picasso', label="Only Picasso", value=FALSE)

output$painting <- renderPlot({
    if (input$picasso) {
      dat <- Paintings[Paintings$Artist=='Pablo Picasso',]
    } else {
      dat <- Paintings
    }
    ...
  })
```

Slider

- Filter by year acquired

```
sliderInput('year', label=h3("Year acquired"),
            min=1930, max=2016, value=c(1950, 2000))
```

Mouse events

- Click, double click, hover, brush

```r
plotOutput('painting',
           click = "plot_click",
           dblclick = "plot_dblclick",
           hover = "plot_hover",
           brush = brushOpts(id = "plot_brush",
                             resetOnNew = TRUE))
```

Click info

- Print mouse events

```
output$click_info <- renderPrint({
    str(input$plot_click)
})

verbatimTextOutput("click_info")
```

Double click

- Observe double click
- Reactive values

```
ranges <- reactiveValues(x = NULL, y = NULL)

observeEvent(input$plot_dblclick, {
    brush <- input$plot_brush
    if (!is.null(brush)) {
      ranges$x <- c(brush$xmin, brush$xmax)
      ranges$y <- c(brush$ymin, brush$ymax)
    } else {
      ranges$x <- NULL
      ranges$y <- NULL
    }
})
```

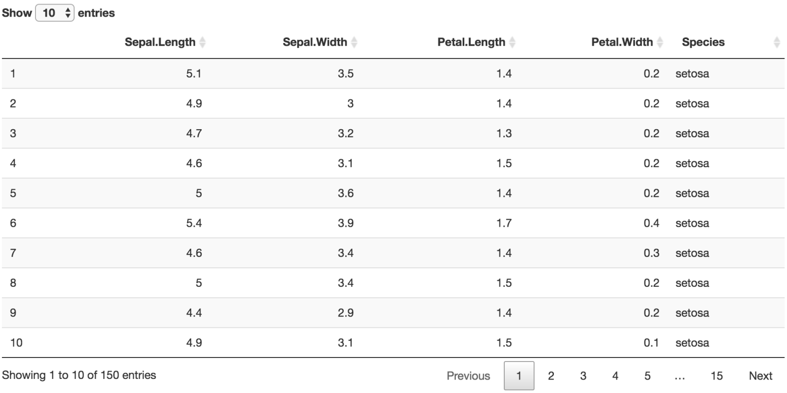
Zoom

- Change plot coordinate system

```
+ coord_cartesian(xlim = ranges$x, ylim = ranges$y)
```

Datatable

- Popular javascript table



Show 10 entries

	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
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa

Showing 1 to 10 of 150 entries

`http://shiny.rstudio.com/gallery/datatables-options.html`

MathJax

- Javascript for displaying LaTex
- Pass xtable output as raw html

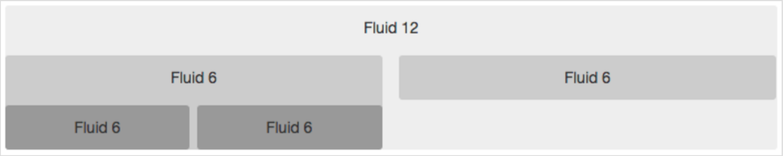
```
output$table <- renderUI({
    M <- print(xtable(M, align=rep("c", ncol(M)+1)),
               floating=FALSE, tabular.environment="array",
               comment=FALSE, print.results=FALSE)
    html <- paste0("$$", M, "$$")
    list(withMathJax(), HTML(html))
})

withMathJax(),
uiOutput('table')
```

http://shiny.rstudio.com/gallery/mathjax.html

Layout

- Fluid grid layout (similar to bootstrap)
- 12 columns every row
- Tabset



Fluid 12

Fluid 6

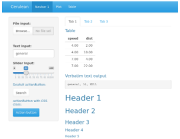
Fluid 6

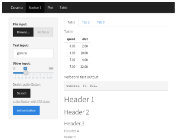
Fluid 6

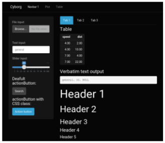
Fluid 6

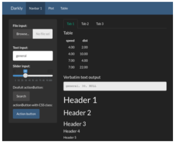
Html

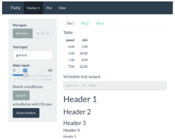
- Customize html and css style files
- Shiny themes

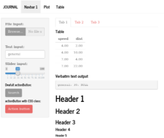


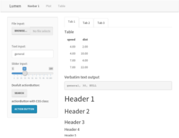


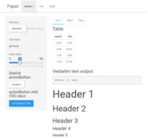












`https://rstudio.github.io/shinythemes/`

Shiny server

- Deploy apps to the interwebs
- `http://shiny.datascience.uci.edu/server/`

Other packages

- Framework for Javascript visualizations

`http://www.htmlwidgets.org/`

Extra

- Cheatsheet

http://shiny.rstudio.com/images/shiny-cheatsheet.pdf

- Gallery with source code

http://shiny.rstudio.com/gallery/

Challenge

- How can you show the name of selected artwork?
- nearPoints()

`http://shiny.rstudio.com/articles/selecting-rows-of-data.html`