Introduction to Shiny

Lingge Li

2/21/2016

What is Shiny

- A framework for building web applications
- Best for interactive data visualization
- Apps for exploratory analysis

http://shiny.datascience.uci.edu/UCIDataScienceInitiative/ClimateActionShiny/
http://shiny.datascience.uci.edu/uciMetropolitanFutures/employment_centers/

Server and UI

- ui.R has everything you see

input widgets, plots, tables. . .

- server.R does the work

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

})

shinyUI(fluidPage(

))
```

Input

- Input handled by specific widgets
- Each input has an id
- Access input value with input$id

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

Output

- Output rendered in server with output$id
- Then displayed in ui

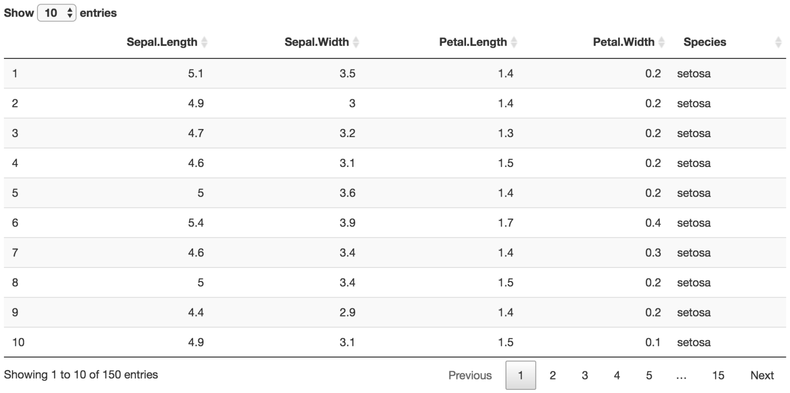
```r
library(shiny)
shinyServer(function(input, output) {
  output$histogram <- renderPlot({
    hist(faithful$eruptions)
  })
})

shinyUI(fluidPage(
  plotOutput(outputId='histogram')
))
```

Several types of output

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

Datatable

- 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

Reactive environment

- Triggered when input changes
- Output changes accordingly
- Rendering functions reactive

Example

```
library(shiny)
library(ggplot2)

shinyServer(function(input, output) {
  # read dataset
  imdb <- read.csv("~/Downloads/movie_metadata.csv")
  # subset nicolas cage movies
  cage <- imdb[imdb$actor_1_name == 'Nicolas Cage', ]
  cage$actor_1_name <- as.character(cage$actor_1_name)
  # create plot
  output$movies <- renderPlot({
    ggplot(data=cage, aes(x=title_year, y=imdb_score, labe
      geom_point(alpha=0.5) +
      geom_text(fontface='italic', size=6, vjust=1, nudge_
      labs(x='Year', y='IMDB Score') +
      geom_smooth()
  })
})
```

Example

```r
shinyUI(fluidPage(
  plotOutput(outputId='movies')
))
```

Add widget

```
shinyUI(fluidPage(
  plotOutput(outputId='movies'),
  selectInput('color', label='Color',
              choices=list('black', 'red', 'blue'),
              selected='black')
))
```

Change color

- Set colour to input$color

```
output$movies <- renderPlot({
  ggplot(data=cage, aes(x=title_year, y=imdb_score, label=n
    geom_point(alpha=0.5, colour=input$color) +
    geom_text(fontface='italic', size=6, vjust=1, nudge_y=0
    labs(x='Year', y='IMDB Score') +
    geom_smooth()
})
```

Slider

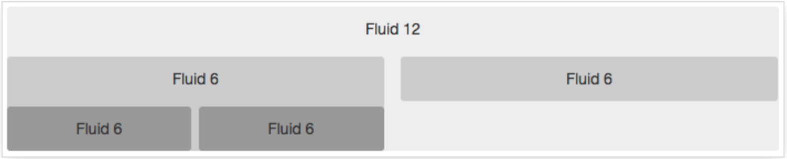
```r
shinyUI(fluidPage(
  plotOutput(outputId='movies'),
  selectInput('color', label='Color',
              choices=list('black', 'red', 'blue'),
              selected='black'),
  sliderInput('year', label="Year",
              min=1980, max=2016, value=c(1980, 2016))
))
```

Filter by input$year

```
output$movies <- renderPlot({
  temp <- cage[cage$title_year >= input$year[1] & cage$titl
  ggplot(data=temp, aes(x=title_year, y=imdb_score, label=m
    geom_point(alpha=0.5, colour=input$color) +
    geom_text(fontface='italic', size=6, vjust=1, nudge_y=0
    labs(x='Year', y='IMDB Score') +
    geom_smooth()
})
```

Layout

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

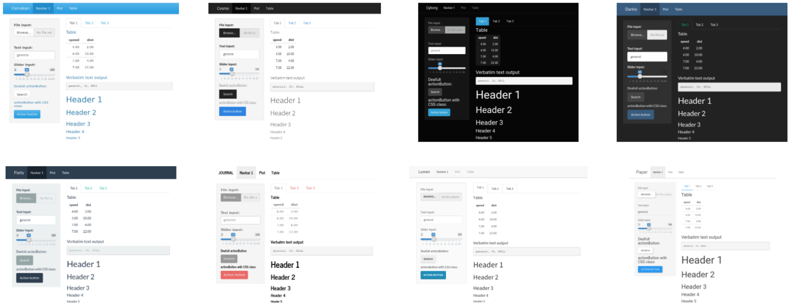


Rows and columns

```
shinyUI(fluidPage(
  fluidRow(align='center',
    plotOutput(outputId='movies')
  ),
  fluidRow(
    column(3, offset=1,
           selectInput('color', label='Color',
                       choices=list('black', 'red', 'blue')
                       selected='black')),
    column(4,
           sliderInput('year', label="Year",
                       min=1980, max=2016, value=c(1980, 20
    )
))
```

Html

- Customize html and css style files
- Shiny themes



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

Interactive documents

- Embed interactive plots in markdown

```
---
title: "Nicolas Cage Movies"
output: html_document
runtime: shiny
---
```

`http://rmarkdown.rstudio.com/authoring_shiny.html`

Shiny server

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

Other packages

- Widgets for Javascript data visualization

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

Resources

- Tutorial

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

- Cheatsheet

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

- Gallery with source code

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