

Introdução ao Tratamento e Análise de Dados em R

R Shiny

Sérgio Rivero

PPGE-UFPA

22 de maio de 2019



Sumário

- 1 Objetivo da Aula
- 2 O que é o shiny?
- 3 Estrutura de um aplicativo *shiny*
- 4 Layouts do shiny
- 5 Alguns links úteis
- 6 Exercícios



Objetivo da Aula

- 1 Apresentar os elementos básicos do Shiny
- 2 Apresentar alguns casos de aplicação da ferramenta



O que é o shiny?

- O *shiny* é um pacote R que facilita a criação de aplicativos interativos com uma interface web Web direto do R.
- Para baixar o *shiny* usamos nosso velho conhecido *install.packages()*



Algumas características do *shiny*

- Permite aplicativos da Web com apenas algumas linhas de código,
- As aplicações em *shiny* mudam suas saídas instantaneamente conforme os usuários modificam as entradas, sem exigir uma recarga do navegador;
- Interfaces de usuário em shiny podem ser construídas inteiramente usando R;
- Funciona em qualquer ambiente R (Console R, Rgui para Windows ou Mac, ESS, StatET, RStudio, etc.)



Estrutura de um aplicativo *shiny*

Aplicativos *shiny* estão contidos em um único script chamado `app.R`. O script `app.R` reside em um diretório (por exemplo, `newdir/`) e o aplicativo pode ser executado com `runApp("newdir")`.

`app.R` tem três componentes:

- 1 um objeto de interface do usuário
- 2 uma função de servidor
- 3 uma chamada para a função `shinyApp`

<https://shiny.rstudio.com/images/shiny-cheatsheet.pdf>



Os Objetos básicos da app *shiny*

O objeto *ui* da interface com o usuário controla o layout e a aparência do seu aplicativo. A função *server* contém as instruções que seu computador precisa para criar seu aplicativo. Finalmente, a função *shinyApp* cria objetos do *shiny* a partir de um par de interface do usuário/servidor explícito.



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



Estrutura de um arquivo de aplicação shiny

FACECON
Faculdade de Ciências Econômicas



Os Objetos básicos da app *shiny*

Um arquivo de aplicação em *shiny*, então, conterà 3 partes (Figura 1). Uma primeira parte é a descrição da estrutura da página a ser publicada, bem como com os elementos da interface do usuário (ui). Este componentes terão os elementos de entrada e também indicarão onde ficariam os gráficos, tabelas e textos que resultariam nas **reações** as mudanças nos valores que estiverem conectados com estas saídas.

Add inputs to the UI with ***Input()** functions

Add outputs with ***Output()** functions

Tell server how to render outputs with R in the server function. To do this:

1. Refer to outputs with **output\$<id>**
2. Refer to inputs with **input\$<id>**
3. Wrap code in a **render*()** function before saving to output

```
library(shiny)

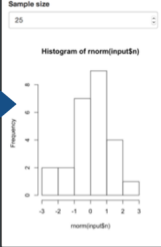
ui <- fluidPage(
  numericInput(inputId = "n",
    "Sample size", value = 25),
  plotOutput(outputId = "hist")
)

server <- function(input, output) {
  output$hist <- renderPlot({
    hist(rnorm(input$n))
  })
}

shinyApp(ui = ui, server = server)
```

Sample size
25

Histogram of rnorm(input\$n)

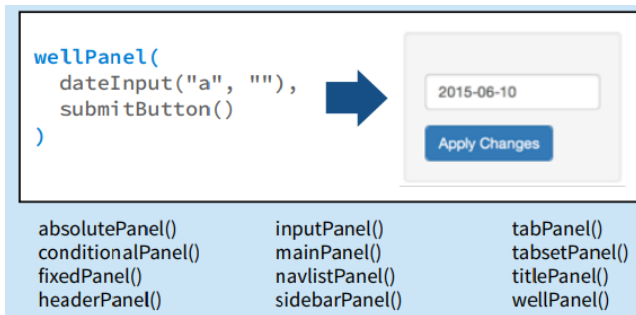




Entrada, servidor e aplicação em detalhe

Layouts do shiny

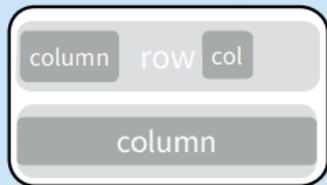
A interface de usuário do shiny permite uma quantidade razoável de opções de layout, podendo também combinar conjuntos diferentes de elementos em entradas únicas. Abaixo (Figura 3) vemos o exemplo de um painel com um campo de entrada tipo data e um botão para incluir o valor da variável.



Um painel composto e as diversas funções de painéis compostos



fluidRow()

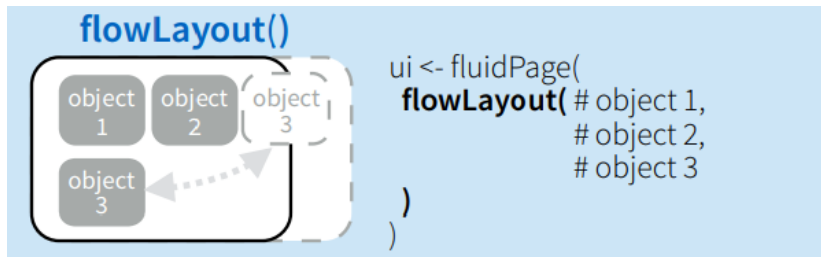


```
ui <- fluidPage(  
  fluidRow(column(width = 4),  
            column(width = 2, offset = 3)),  
  fluidRow(column(width = 12))  
)
```

Exemplo da função fluidRow()



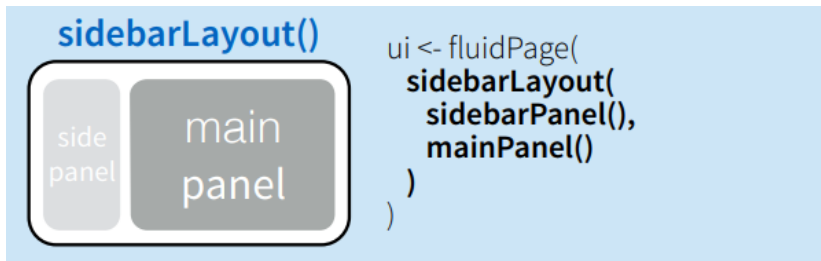
flowLayout



Exemplo da função flowLayout()



sidebarLayout



Exemplo da função sidebarLayout()



Painéis no *shiny*



Layer tabPanels on top of each other,
and navigate between them, with:

```
ui <- fluidPage( tabsetPanel(  
  tabPanel("tab 1", "contents"),  
  tabPanel("tab 2", "contents"),  
  tabPanel("tab 3", "contents")))
```

```
ui <- fluidPage( navlistPanel(  
  tabPanel("tab 1", "contents"),  
  tabPanel("tab 2", "contents"),  
  tabPanel("tab 3", "contents")))
```

```
ui <- navbarPage(title = "Page",  
  tabPanel("tab 1", "contents"),  
  tabPanel("tab 2", "contents"),  
  tabPanel("tab 3", "contents"))
```



Alguns Tipos de Entradas

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

passwordInput(inputId, label, value)

- ☒ Choice A
- ☐ Choice B
- ☐ Choice C

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

Choice 1 ▲

Choice 1

Choice 2

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)



Funções de Renderização e Saída

Outputs - render*() and *Output() functions work together to add R output to the UI

DT::renderDataTable(expr,
options, callback, escape,
env, quoted)



dataTableOutput(outputId, icon, ...)



renderImage(expr, env, quoted, deleteFile)

imageOutput(outputId, width, height, click,
dblclick, hover, hoverDelay, hoverDelayType,
brush, clickId, hoverId, inline)



renderPlot(expr, width, height, res, ..., env,
quoted, func)

plotOutput(outputId, width, height, click,
dblclick, hover, hoverDelay, hoverDelayType,
brush, clickId, hoverId, inline)

1 Value: TRUE
2 Value: FALSE
3 Value: TRUE
4 Value: FALSE
5 Value: TRUE
6 Value: FALSE
7 Value: TRUE
8 Value: FALSE
9 Value: TRUE
10 Value: FALSE

renderPrint(expr, env, quoted, func,
width)

verbatimTextOutput(outputId)

renderTable(expr, ..., env, quoted, func)

tableOutput(outputId)

foo

renderText(expr, env, quoted, func)

textOutput(outputId, container, inline)



renderUI(expr, env, quoted, func)

uiOutput(outputId, inline, container, ...)
& **htmlOutput**(outputId, inline, container, ...)



Exemplos

https:

[//shiny.rstudio.com/gallery/retirement-simulation.html](https://shiny.rstudio.com/gallery/retirement-simulation.html)

<https://shiny.rstudio.com/gallery/tabsets.html>



Alguns links úteis

<https://rstudio.github.io/shiny/tutorial/>

<https://shiny.rstudio.com/tutorial>

<https://shiny.rstudio.com/>

<https://www.rstudio.com/products/shiny/shiny-user-showcase/>

<https://www.showmeshiny.com/category/topics/finance/>

<https://shiny.rstudio.com/images/shiny-cheatsheet.pdf>



Exercícios

Exercícios

