



利用R语言进行交互 ——玩玩shiny

第九届中国R语言会议（广州）暨华南地区数据科学会议

中山大学

R Square成员: 蒋宇康 慕聪



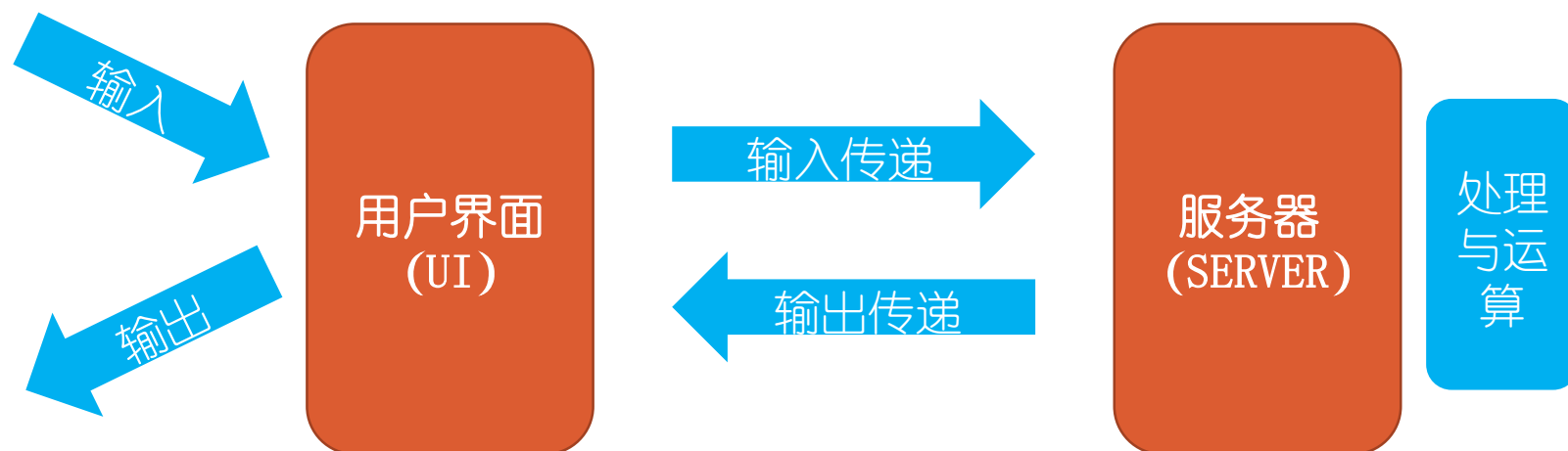
目录

- ① 简介
- ② UI布局
- ③ 输入与输出
- ④ 实战演练

R中的shiny包



- 可以利用R语言**轻松快速**地开发交互式web应用



最简单的shiny



```
library(shiny)
shinyUI(pageWithSidebar(
  headerPanel("利用R语言进行交互——玩玩shiny"),
  sidebarPanel(),
  mainPanel()
))
```

ui.R

```
library(shiny)
shinyServer(function(input, output) {
})
```

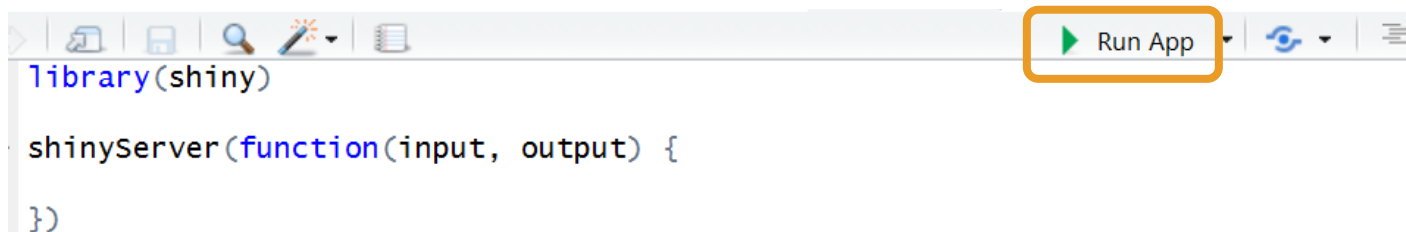
server.
R

三种运行方式



- ① `library(shiny)`
`runApp("D:/myapp")` #两个文件存到D盘的myapp文件夹中

- ② 点击RStudio脚本文件右上角的Run App



- ③

```
ui <- shinyUI(pageWithSidebar(  
  headerPanel("利用R语言进行交互--玩玩shiny"),  
  sidebarPanel(),  
  mainPanel()  
)  
  
server <- shinyServer(function(input, output) {  
  })  
  
shinyApp(ui, server)
```

输出：



利用R语言进行交互——玩玩shiny



UI布局

- 对于页面的布局，推荐使用shinydashboard包。
- 对全部，先使用dashboardPage()将所有内容扩起。
- 标题： dashboardHeader(title= " ")
- 侧边栏： dashboardSidebar()
- 主体： dashboardBody()



UI布局

皮肤颜色选择: Blue、Black、Purple、Green、Red、Yellow

```
ui <- dashboardPage(skin = 'green',  
  dashboardHeader(title = '利用R语言进行交互'),  
  dashboardSidebar(  
    sidebarMenu(  
      menuItem('布局', tabName = 'layout', icon = icon('th')),  
      menuItem('表格', tabName = 'table', icon = icon('table')),  
      menuItem('图', tabName = 'plot1', icon = icon('picture-o')),  
      menuItem('更美的图', tabName = 'plot2', icon = icon('picture-o')),  
      menuItem('再美一些的图', tabName = 'plot3', icon = icon('picture-o')),  
      menuItem('最美的图', tabName = 'plot4', icon = icon('picture-o'),  
        badgeLabel = "Best!", badgeColor = "yellow")  
    ),  
    dashboardBody(  
      tabItems(  
        tabItem(tabName = 'layout',
```

侧边栏小标签名称

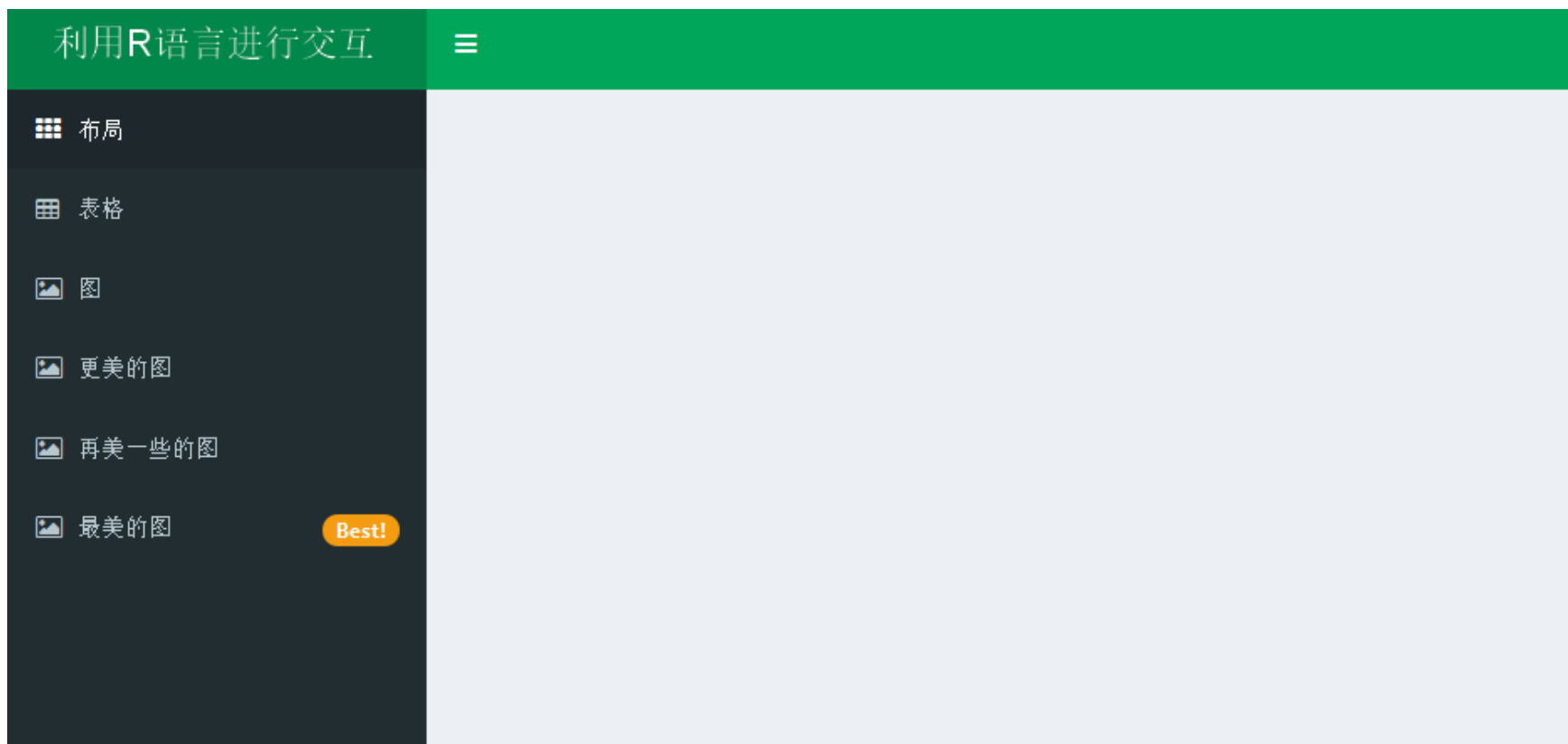
侧边栏图标

图标样式请参考:

<http://fontawesome.io/icons/>

<http://getbootstrap.com/components/#glyphicons>

UI布局





```
dashboardSidebar(  
  sidebarMenu(  
    menuItem('布局', tabName = 'layout', icon = icon('th'),  
      menuItem("输入", tabName = 'input1', icon = icon('sign-in')),  
      menuItem("输出", tabName = 'output1', icon = icon('sign-out')) ),  
    menuItem('表格', tabName = 'table', icon = icon('table')),  
    menuItem('图', tabName = 'plot1', icon = icon('picture-o')),  
    menuItem('更美的图', tabName = 'plot2', icon = icon('picture-o')),  
    menuItem('再美一些的图', tabName = 'plot3', icon = icon('picture-o')),  
    menuItem('最美的图', tabName = 'plot4', icon = icon('picture-o'),  
      badgeLabel = "Best!", badgeColor = "yellow")  
  ),  
)
```



输入



`sliderInput()`

滑块条输入

`numericInput()`

数值输入

`selectInput()`

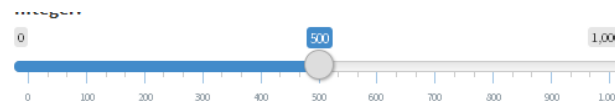
选项列表输入

`radioButtons()`

选项卡输入

`textInput()`

文字输入



选项1

选项1

选项2

请选择:

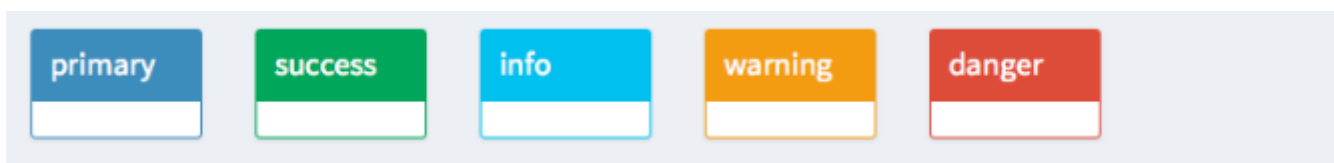
☐ 选项1

☐ 选项2

小区块颜色



使用 “status = ” 控制颜色：



使用 “color = ” 控制颜色：





UI布局

按行排列, 按列排布需要用: `column()`

```
dashboardBody(  
  tabItems(  
    tabItem(tabName = 'input1',  
      fluidRow(  
        box(  
          title = "滑动输入", width = 6, status = "primary",  
          sliderInput("integer", "Integer:",  
            min=0, max=1000, value=500)  
        ),  
        box(  
          title = "手动输入", width = 6, status = "warning",  
          numericInput("integer", "Integer:", 10)  
        )  
      ),  
    fluidRow(  
      box(  
        title = "选项卡", width = 4, solidHeader = TRUE, status = "success",  
        radioButtons('select', "请选择:",  
          list("选项1" = "select1",  
              "选项2" = "select2"))  
      ),  
      box(  
        title = "选项列表", width = 4, solidHeader = TRUE, status = "info",  
        selectInput('select', "请选择:",  
          choices = c("选项1", "选项2", "选项3"))  
      ),  
      box(  
        title = "文字输入", width = 4, solidHeader = TRUE, background = 'maroon',  
        textInput("text", "请输入: ", value = "输入...")  
      )  
    )  
  )  
)
```

小区块主色调

标题填充色

背景填充色

代表一个小区块

`box()`

UI布局



利用R语言进行交互



布局

输入

输出

表格

图

更美的图

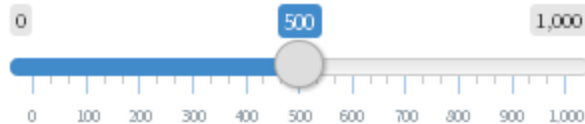
再美一些的图

最美的图

Best!

滑动输入

Integer:



手动输入

Integer:

10

选项卡

请选择:

☒ 选项1

☐ 选项2

选项列表

请选择:

选项1



文字输入

请输入:

输入...

输出



- 文本输出

Server : **renderPrint**

UI : **verbatimTextOutput & textOutput**

- 表格输出

Server : **renderTable**

UI : **tableOutput**

- 图片输出

Server : **renderPlot**

UI : **plotOutput**



输出(以iris数据集为例)

UI :

```
tabItem(tabName = 'output1',  
  fluidRow(  
    box(  
      title = "结果输出", width = 6, solidHeader = TRUE, status = "primary",  
      verbatimTextOutput("text.iris")  
    ),  
    box(  
      title = "Plot输出", width = 6, solidHeader = TRUE, status = "success",  
      plotOutput("plot.iris", height = 250)  
    ),  
    box(  
      title = "表格输出", width = 12, solidHeader = TRUE, status = "warning",  
      tableOutput("table.iris")  
    )  
  ),  
)
```

Server

```
server <- function(input, output) {  
  output$text.iris = renderPrint({  
    summary(iris) })  
  output$table.iris = renderTable({  
    iris })  
  output$plot.iris = renderPlot({  
    plot(iris) })  
}
```


输出



利用R语言进行交互



布局

输入

输出

表格

图

更美的图

再美一些的图

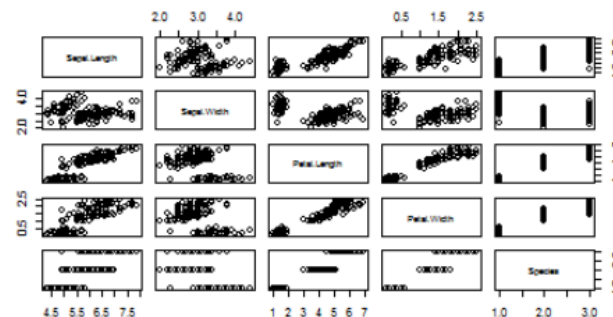
最美的图

Best!

结果输出

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
Min. :4.300	Min. :2.000	Min. :1.000	Min. :0.100	setosa
1st Qu.:5.100	1st Qu.:2.800	1st Qu.:1.600	1st Qu.:0.300	versicolor
Median :5.800	Median :3.000	Median :4.350	Median :1.300	virginica
Mean :5.843	Mean :3.057	Mean :3.758	Mean :1.199	
3rd Qu.:6.400	3rd Qu.:3.300	3rd Qu.:5.100	3rd Qu.:1.800	
Max. :7.900	Max. :4.400	Max. :6.900	Max. :2.500	

表格输出



Plot输出

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.10	3.50	1.40	0.20	setosa
4.90	3.00	1.40	0.20	setosa
4.70	3.20	1.30	0.20	setosa
4.60	3.10	1.50	0.20	setosa



添加标签页

```
tabItem(tabName = 'table',
  tabsetPanel(
    tabPanel("普通表格",
      tableOutput("table.hot")
    ),
    tabPanel("交互表格",
      box(
        title = tagList(icon("sign-in"), "输入"),
        width = 4, background = "black",
        radioButtons('select1', "请选择:",
          list("按频道" = "1",
            "按时间" = "2")
        ),
        selectInput('program', "请选择频道:",
          choices = hot.pro),
        selectInput('time.sel', "请选择时间:",
          choices = time),
        submitButton("更新视图")
      ),
      tabBox(
        title = tagList(icon("table"), "表格展现"),
        width = 8,
        tabPanel("使用View()",
          tableOutput("output.table1")
        ),
        tabPanel("使用DT包",
          dataTableOutput("output.table2")
        )
      )
    )
  ),
),
```

标签页

利用R语言进行交互



布局

收视数据表格

普通表格

交互表格

program 00:30 channel 00:30_audie



```
tabsetPanel(  
  tabPanel("普通表格",  
    tableOutput("table.hot")  
  ),  
  
  tabPanel("交互表格",  
    box(  
      title = tagList(icon("sign-in"), "输入"),  
      width = 4, background = "black",  
      radioButtons('select1', "请选择:",  
        list("按频道" = "1",  
              "按时间" = "2")  
      ),  
      selectInput('program', "请选择频道:",  
        choices = hot.pro),  
      selectInput('time.sel', "请选择时间:",  
        choices = time),  
      submitButton("更新视图")  
    )  
  )  
)
```

➡ 输入

请选择:

☐ 按频道

☒ 按时间

请选择频道:

3D频道

请选择时间:

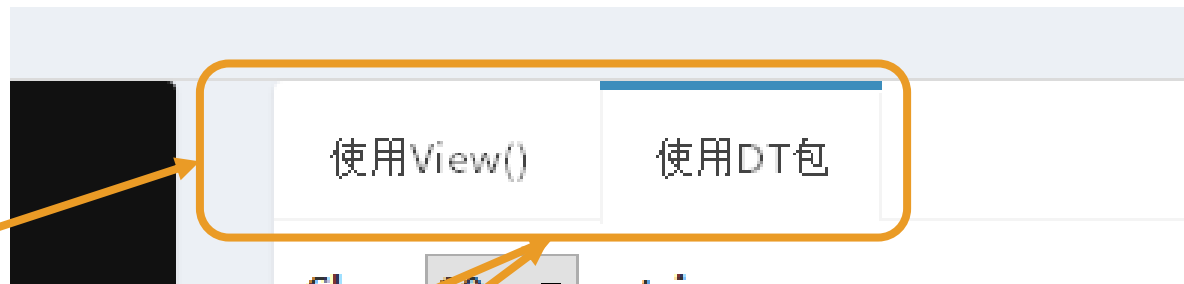
00:30

更新视图

```
title = tagList(icon("sign-in"), "输入"),  
width = 4, background = "black",  
radioButtons('select1', "请选择:",  
  list("按频道" = "1",  
        "按时间" = "2")  
)  
,  
selectInput('program', "请选择频道:",  
  choices = hot.pro),  
selectInput('time.sel', "请选择时间:",  
  choices = time),  
submitButton("更新视图")  
)
```



Box中的标签页



```
tabBox(  
  title = tagList(icon("table"), "表格展现"),  
  width = 8,  
  tabPanel("使用View()",  
    tableOutput("output.table1")  
  ),  
  tabPanel("使用DT包",  
    dataTableOutput("output.table2")  
  )  
)
```

标签页



利用R语言进行交互



布局

收视数据表格

图

更美的图

再美一些的图

最美的图

Best!

普通表格

交互表格

➡ 输入

请选择:

☐ 按频道

☒ 按时间

请选择频道:

3D频道

请选择时间:

00:30

更新视图

使用View()

使用DT包

表格展现

Show 10 entries

Search:

	时间	节目	收视率
1	00:30		
2	01:00	特别节目(3)	0.000643086816720257
3	01:30	泉水之城-瑞士圣莫里茨	0.000789577575996842
4	02:00	泉水之城-瑞士圣莫里茨	0.000956937799043062
5	02:30		
6	03:00		
7	03:30	梦想-家2012群星新春大联欢(0.00105932203389831
8	04:00		
9	04:30	NULL	0.00148588410104012
10	05:00	NULL	0.0013550135501355

Showing 1 to 10 of 48 entries

Previous

1

2

3

4

5

Next



● 实战演练

——以收视数据为例