BUILDING DASHBOARDS WITH SHINYDASHBOARD

Reactive expression refresher

Lucy D'Agostino McGowan
Graduate Researcher in Biostatistics at Vanderbilt University

Inputs

- actionButton()
- checkboxInput()
- dateInput()
- numericInput()
- radioButtons()
- selectInput()
- sliderInput()
- textInput()

Inputs

selectInput()

- inputId
- label
- choices

```
selectInput(
  inputId = "favorite_character",
  label = "Favorite Character",
  choices = c("Leia", "Luke", "Han")
  )
```

Render functions

- renderPrint()
- renderText()
- renderTable()
- renderDataTable()
- renderPlot()
- renderImage()
- renderUI()

Render Functions

```
server <- function(input, output) {

  output$printed_object <- renderPrint({
    "print me"
  })
}
```

Output Functions

- renderPrint() -- verbatimTextOutput(), textOutput()
- renderText() -- textOutput()
- renderTable() -- tableOutput()
- renderDataTable() -- dataTableOutput()
- renderPlot() -- plotOutput()
- renderImage() -- imageOutput()
- renderUI() -- uiOutput()

Output Functions

- renderPrint() -- verbatimTextOutput(), textOutput()
- render**Text**() -- **text**Output()
- render**Table**() -- **table**Output()
- render**DataTable**() -- **dataTable**Output()
- render**Plot**() -- **plot**Output()
- render**Image**() -- **image**Output()
- render**UI**() -- **ui**Output()

Output Functions

textOutput()

- outputId

```
textOutput("printed_object")
```

BUILDING DASHBOARDS WITH SHINYDASHBOARD

Let's practice!

BUILDING DASHBOARDS WITH SHINYDASHBOARD

Server-side dynamic how-to

Lucy D'Agostino McGowan
Graduate Researcher in Biostatistics at Vanderbilt University

Read in real time data

```
server <- function(input, output, session) {
  reactive_data <- reactiveFileReader(
    intervalMillis = 1000,
    session = session,
    filePath = "file.csv",
    readFunc = read.csv
  )
}
```

Read in real time data

reactiveFileReader()

- intervalMillis
- session
- filePath
- readFunc

```
server <- function(input, output, session) {
  reactive_data <- reactiveFileReader(
    intervalMillis = 1000,
    session = session,
    filePath = "file.csv",
    readFunc = read.csv
  )
}
```

Read in real time data

```
server <- function(input, output, session) {
 ...
 output$printed_data <- renderPrint({
   reactive_data()
   })
}
```

BUILDING DASHBOARDS WITH SHINYDASHBOARD

Let's build a dynamic dashboard!

BUILDING DASHBOARDS WITH SHINYDASHBOARD

Optimizing performance

Lucy D'Agostino McGowan
Graduate Researcher in Biostatistics at Vanderbilt University

Outside the server function

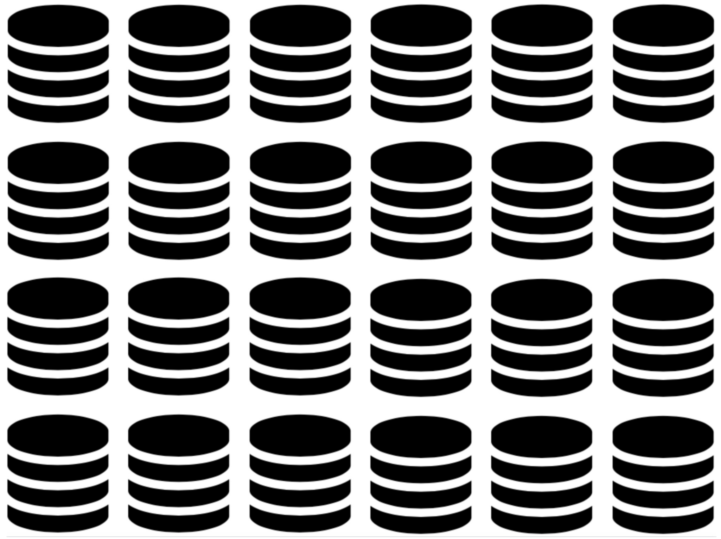
```
load("starwars.Rda")
server <- function(input, output) {
}
```

Inside the server function

```
server <- function(input, output) {
  load("starwars.Rda")
}
```

Inside the server function, inside a reactive object

```
server <- function(input, output) {
  output$my_plot <- renderPlot({
    load("starwars.Rda")
    starwars_sub <- starwars[starwars$height <= input$height, ]
    hist(starwars_sub$height)
  })
}
```



BUILDING DASHBOARDS WITH SHINYDASHBOARD

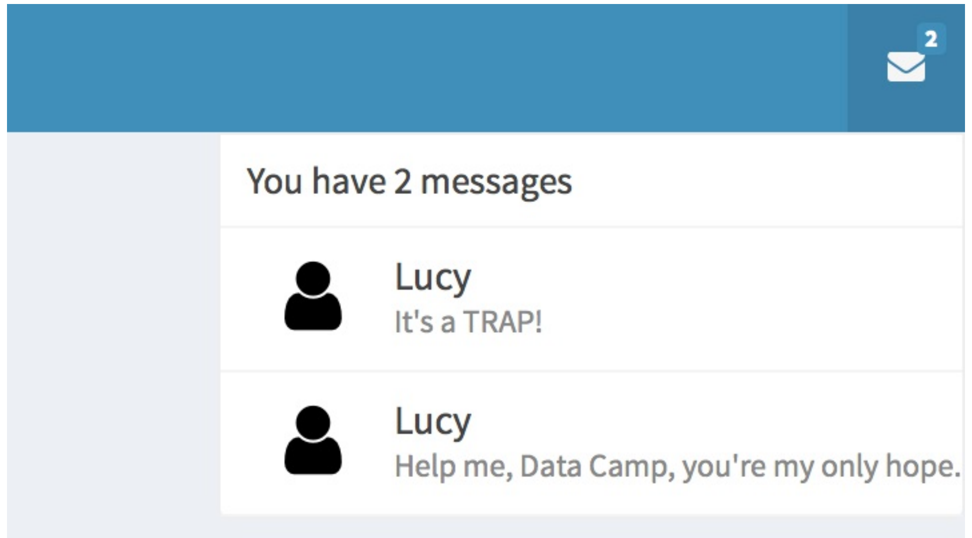
May the force be with you!

BUILDING DASHBOARDS WITH SHINYDASHBOARD

UI dynamic how-to

Lucy D'Agostino McGowan
Graduate Researcher in Biostatistics at Vanderbilt University

Dynamic Dropdown Menu



2

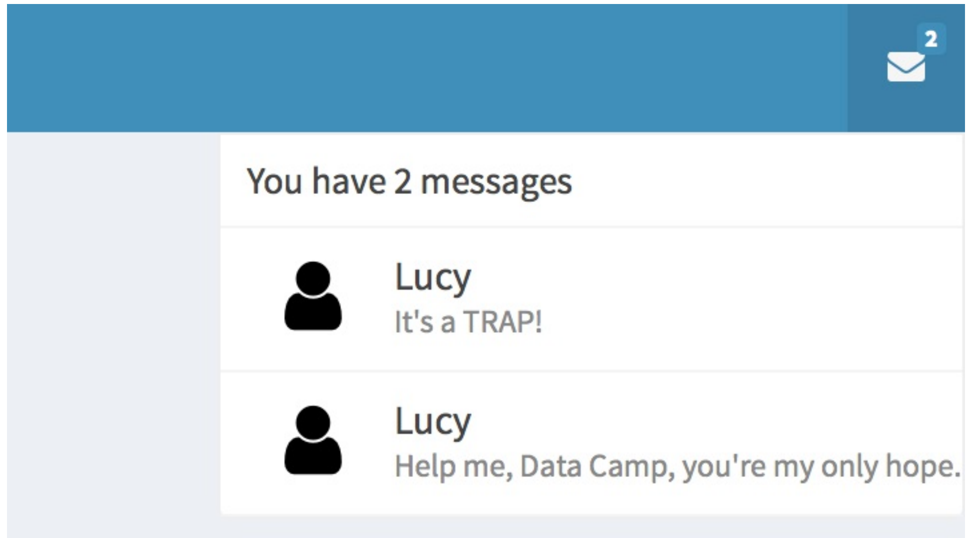
You have 2 messages

Lucy
It's a TRAP!

Lucy
Help me, Data Camp, you're my only hope.

```
dropdownMenu(
  type = "messages",
    messageItem(
      from = "Lucy",
      message = "It's a TRAP!"
    ),
    messageItem(
      from = "Lucy",
      message = "Help me, Data Camp, you're my only hope."
    )
)
```

Dynamic Dropdown Menu



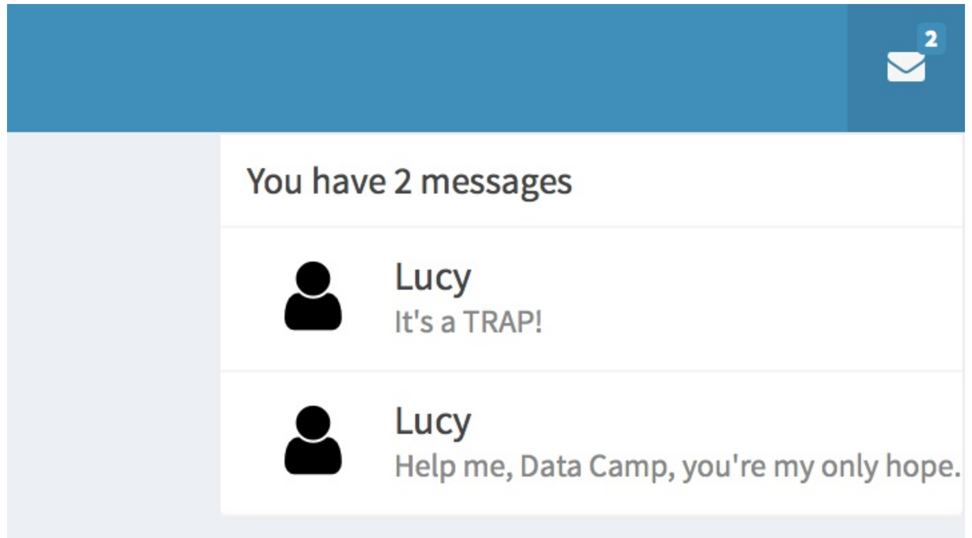
You have 2 messages

Lucy
It's a TRAP!

Lucy
Help me, Data Camp, you're my only hope.

```
msg_df
#>   from                                  message
#> 1 Lucy                             It's a TRAP!
#> 2 Lucy Help me, Data Camp, you're my only hope.
```

Dynamic Dropdown Menu



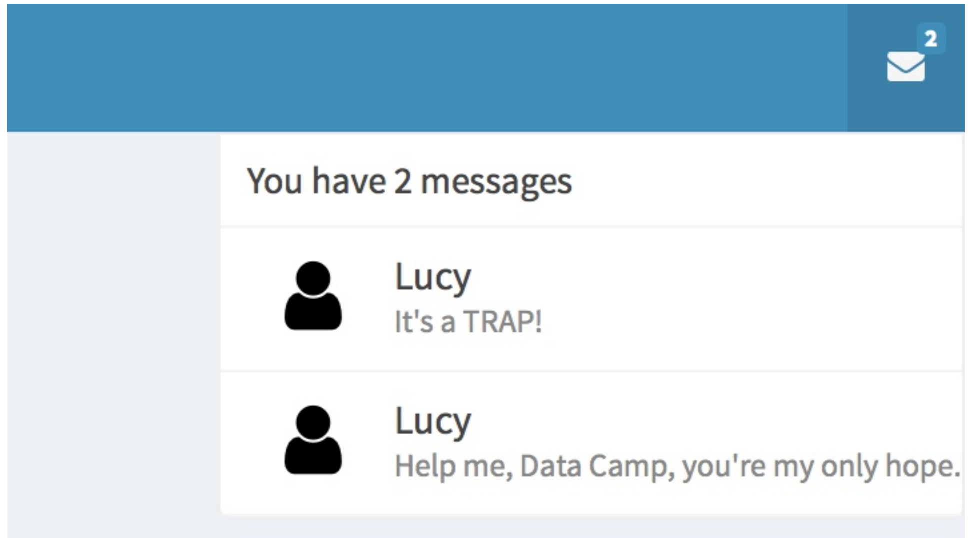
You have 2 messages

Lucy
It's a TRAP!

Lucy
Help me, Data Camp, you're my only hope.

```
messages <- apply(data, 1, function(row) {
  messageItem(from = row[["from"]],
              message = row[["message"]])
})
dropdownMenu(type = "message", .list = messages)
```

Dynamic Dropdown Menu



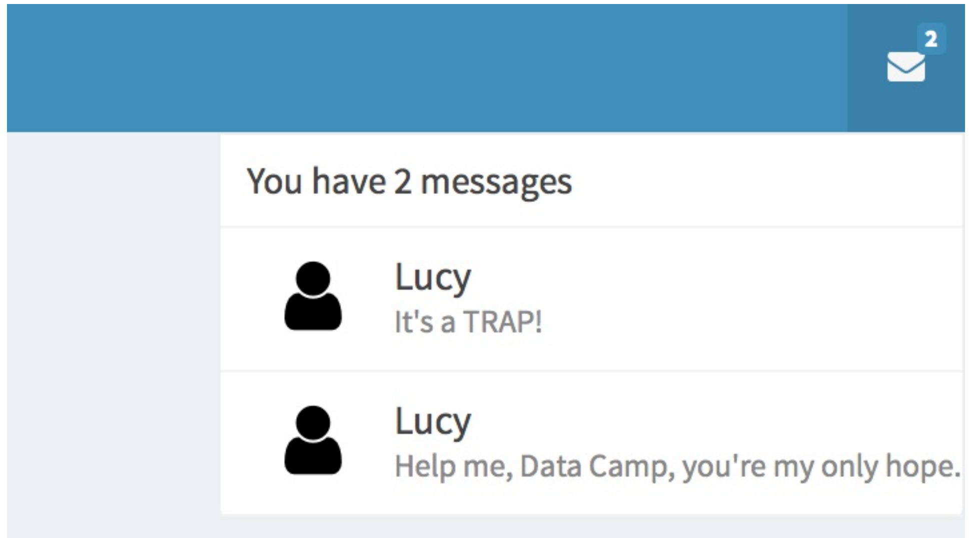
You have 2 messages

Lucy
It's a TRAP!

Lucy
Help me, Data Camp, you're my only hope.

```
server <- function(input, output) {
  output$msg_menu <- renderMenu({
    messages <- apply(data, 1, function(row) {
    messageItem(from = row[["from"]],
                message = row[["message"]])
    })
    dropdownMenu(type = "message", .list = messages)
  })
}
```

Dynamic Dropdown Menu



You have 2 messages

Lucy
It's a TRAP!

Lucy
Help me, Data Camp, you're my only hope.

```
header <- dashboardHeader(dropdownMenuOutput("msg_menu"))
```

Dynamic boxes

- renderInfoBox() -- infoBoxOutput()
- renderValueBox() -- valueBoxOutput()

BUILDING DASHBOARDS WITH SHINYDASHBOARD

You give it a try!