shiny tutorial

Bharvi Dhall

1/23/2020

This is a Shiny web application. You can run the application by clicking

the 'Run App' button above.

Find out more about building applications with Shiny here:

http://shiny.rstudio.com/

INTRODUCTION

```
library(shiny)

# Shiny has two parts - UI and Server operations
#UI takes care of the visual appearance of the page
#Server tells you how the app works

# Define UI for application that draws a histogram
ui <- fluidPage(
    selectInput("dataset", label = "Dataset", choices = ls("package:datasets")),
    verbatimTextOutput("summary"),
    tableOutput("table")
)

## fluidPage gives the layout to the page. selectInput allows the user to supply the input value
## the verbatimTextOutput displays the output code and the tableOutput displays the output table.


# Define server logic required to draw a histogram
server <- function(input, output, session) {
    output$summary <- renderPrint({
        dataset <- get(input$dataset, "package:datasets")        #takes input and prints summary
        summary(dataset)
    })

    output$table <- renderTable({
        dataset <- get(input$dataset, "package:datasets")        #takes input and prints table
        dataset
```

```
    })
}
# Run the application
shinyApp(ui = ui, server = server)
```

```
##If we notice the about code has a duplicate statement get(input$dataset, "package:datasets").
##To get rid of this we can write the code inside reactive ({......}) and then assign it to a variable
## This executes code for the first time and caches the result until its updated.

# The user Interface
ui <- fluidPage(
  selectInput("dataset", label = "Dataset", choices = ls("package:datasets")),
  verbatimTextOutput("summary"),
  tableOutput("table")
)

# The server functions
server <- function(input, output, session) {
  dataset <- reactive ({
    get(input$dataset, "package:datasets")
  })

output$summary <- renderPrint({
  summary(dataset())
})

output$table <- renderTable({
  dataset()
})

}
```