etsinf

Escola Tècnica
Superior d'Enginyeria
Informàtica

Introduction to Rstudio Shiny

Cèsar Ferri Ramírez
cferri@disc.upv.es
D235 DSIC

Shiny

- Open-Sourced by RStudio 11/2012 on CRAN
- New model for web-accessible R code
- Able to generate basic web UIs
- Uses web sockets
- Built on a “Reactive Programming” model
- Entirely extensible
- Custom inputs and outputs

Shiny

- Shiny applications have two components:
  - a user interface definition *(Frontend)*
    - *UI.R*
  - server script *(Backend)*
    - *Server.R*
- shinyApp - combines ui and server into a functioning app.
  - Wrap with runApp() if calling from a sourced script or inside a function.

Shiny Example

- ui.R

```
      library(shiny)
# Define UI for miles per gallon application
shinyUI(pageWithSidebar(
  # Application title
  headerPanel("Miles Per Gallon"),
  sidebarPanel(),
  mainPanel()
))
```

- The three functions *headerPanel*, *sidebarPanel*, and *mainPanel* define the various regions of the user-interface.
- The application will be called “Miles Per Gallon”
- http://shiny.rstudio.com/articles/build.html

Shiny Example

- server.R

```
library(shiny)

# Define server logic required to plot various variables against mpg
shinyServer(function(input, output) {

})
```

- The server function is empty
- To create the app

```
> library(shiny)
> runApp("~/shinyapp")
```

Inputs and Outputs

- We use the *mtcars* data (R datasets package)
- Box-plot that explores the relationship between miles-per-gallon (MPG) and three other variables (Cylinders, Transmission, and Gears).

```
library(shiny)

# Define UI for miles per gallon application
shinyUI(pageWithSidebar(
  # Application title
  headerPanel("Miles Per Gallon"),
  # Sidebar with controls to select the variable to plot against mpg
  # and to specify whether outliers should be included
  sidebarPanel(
    selectInput("variable", "Variable:",
                list("Cylinders" = "cyl",
                     "Transmission" = "am",
                     "Gears" = "gear")),
    checkboxInput("outliers", "Show outliers", FALSE)
  ),
  mainPanel()
))
```

Server Script

- The server-side of the application which will accept inputs and compute outputs.
  - Accessing input using slots on the input object and generating output by assigning to slots on the output object.
  - Initializing data at startup that can be accessed throughout the lifetime of the application.
  - Using a reactive expression to compute a value shared by more than one output.

Server Script

```
library(shiny)
library(datasets)

# We tweak the "am" field to have nicer factor labels. Since this doesn't
# rely on any user inputs we can do this once at startup and then use the
# value throughout the lifetime of the application
mpgData <- mtcars
mpgData$am <- factor(mpgData$am, labels = c("Automatic", "Manual"))

# Define server logic required to plot various variables against mpg
shinyServer(function(input, output) {

  # Compute the formula text in a reactive expression since it is
  # shared by the output$caption and output$mpgPlot expressions
  formulaText <- reactive({
    paste("mpg ~", input$variable)
  })

  # Return the formula text for printing as a caption
  output$caption <- renderText({
    formulaText()
  })

  # Generate a plot of the requested variable against mpg and only
  # include outliers if requested
  output$mpgPlot <- renderPlot({
    boxplot(as.formula(formulaText()),
            data = mpgData,
            outline = input$outliers)
  })
})
```

Displaying Outputs

- The server script assigned two output values: *output$caption* and *output$mpgPlot*.
  - To update the user interface to display the output we need to add some elements to the main UI panel.

- We add the caption as an *h3* element and filled in its value using the *textOutput* function
- We also render the plot by calling the *plotOutput* function

Displaying Outputs

```
library(shiny)

# Define UI for miles per gallon application
shinyUI(pageWithSidebar(

  # Application title
  headerPanel("Miles Per Gallon"),

  # Sidebar with controls to select the variable to plot against mpg
  # and to specify whether outliers should be included
  sidebarPanel(
    selectInput("variable", "Variable:",
                list("Cylinders" = "cyl",
                     "Transmission" = "am",
                     "Gears" = "gear")),

    checkboxInput("outliers", "Show outliers", FALSE)
  ),

  # Show the caption and plot of the requested variable against mpg
  mainPanel(
    h3(textOutput("caption")),

    plotOutput("mpgPlot")
  )
))
```

Shinny Apps

- RStudio offers three ways to host your Shiny app as a web page:
  - Shinyapps.io:
    - RStudio's hosting service for Shiny apps
  - Shiny Server
    - It builds a web server designed to host Shiny apps. It's free, open source, and available from Github.
  - Shiny Server Pro
    - If you use Shiny in a for-profit setting, you may want to give yourself the server tools: Password authentification, SSL support, Administrator tools, Priority support...

shinyapps.io

- You can upload apps in shinyapps directly from rstudio
- Or by code: http://shiny.rstudio.com/articles/shinyapps.html

```
install.packages('rsconnect')
library(rsconnect)
rsconnect::setAccountInfo(name="<ACCOUNT>", token="<TOKEN>", secret="<SECRET>")
deployApp(appName = "traffic_local")
```