Intro to Shiny

Shiny is an R package by RStudio that enables you to create web applications with interactive tables and graphs. RStudio has a great tutorial on building and deploying Shiny apps. This tutorial will get you started, but the best place to learn about Shiny is the RStudio website.

## Basics

To make a Shiny app you need to have two .R scripts in the same folder:

* a user-interface (ui) script
* a server script

## ui.R

The ui.R script determines the layout of your app. Let's look at the ui.R for the Hello Shiny example from the RStudio tutorial:

library(shiny)  
  
# Define UI for application that draws a histogram  
shinyUI(fluidPage(  
  
 # Application title  
 titlePanel("Hello Shiny!"),  
  
 # Sidebar with a slider input for the number of bins  
 sidebarLayout(  
 sidebarPanel(  
 sliderInput("bins",  
 "Number of bins:",  
 min = 1,  
 max = 50,  
 value = 30)  
 ),  
  
 # Show a plot of the generated distribution  
 mainPanel(  
 plotOutput("distPlot")  
 )  
 )  
))

## server.R

The server.R script is where you put the code that will build the app.

library(shiny)  
  
# Define server logic required to draw a histogram  
shinyServer(function(input, output) {  
  
 # Expression that generates a histogram. The expression is  
 # wrapped in a call to renderPlot to indicate that:  
 #  
 # 1) It is "reactive" and therefore should be automatically  
 # re-executed when inputs change  
 # 2) Its output type is a plot  
  
 output$distPlot <- renderPlot({  
 x <- faithful[, 2] # Old Faithful Geyser data  
 bins <- seq(min(x), max(x), length.out = input$bins + 1)  
  
 # draw the histogram with the specified number of bins  
 hist(x, breaks = bins, col = 'darkgray', border = 'white')  
 })  
  
})

## Running the app locally

To run an app locally, use the runApp() function. The ui.R and server.R scripts for the **Hello Shiny** app are found in the shiny library, so you can run this app by placing the directory in the first argument:

library(shiny)  
  
folder <- system.file("examples/01\_hello", package="shiny")  
  
runApp(appDir = folder)

## Chicago Air

We'll make a simple app using the the chicago\_air dataset.

**ui.R**

library(shiny)  
  
# Define UI for application that plots time series  
shinyUI(fluidPage(  
  
 # Application title  
 titlePanel("Chicago Air"),  
  
 # Sidebar with a dropdown for selecting parameter  
 sidebarLayout(  
 sidebarPanel(  
 selectInput(inputId = "parameter",  
 label = "Select Parameter:",  
 choices = c("Ozone" = "ozone",  
 "Temperature" = "temp",  
 "Solar Radiation" = "solar"))  
 ),  
  
 # Show a plot of the time series  
 mainPanel(  
 plotOutput("timePlot")  
 )  
 )  
))

**server.R**

library(shiny)  
library(region5air)  
library(ggplot2)  
data(chicago\_air)  
chicago\_air$date <- as.Date(chicago\_air$date)  
  
 # Define server logic required to plot the time series  
shinyServer(function(input, output) {  
   
 output$timePlot <- renderPlot({  
 ggplot(chicago\_air, aes\_string("date", input$parameter)) + geom\_line()   
 })  
})

If we have the ui.R and server.R scripts in a folder called ChicagoApp in our working directory, then we could run the app with

library(shiny)  
runApp("ChicagoApp")

## Exercises

Go to the exercises page for this tutorial and test your comprehension of this material.