Intro to RShiny

Martin Paleico

What is RShiny?

- R package that allows for constructing interactive web application
- Simple to program (with some R knowledge), without needing to know much about HTML
- Good for showcasing paper results, interactive presentations, etc
- Server software for easy hosting also available
- Easy to install, but it also already runs directly on RStudio, like our own instance on rstudio.gwdg.de! (File - New File - Shiny Web App)

File Setup

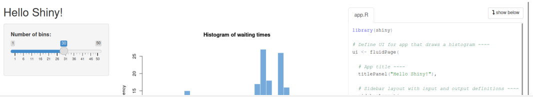
- Either single app.R, or separate ui.R and server.R
- plus an optional global.R
- and an optional www folder where things such as images can be stored
- all stored in a single folder, where the name of the folder will be the name of the app
- Start your app with `runApp("folder−name")`, will open a new tab in your default web browser
- Apps can be started in "showcase" mode which displays the app’s code to the end-user

```
runApp("MyApp", display.mode = "showcase")
#or have DESCRIPTION and Readme.md files
```

Code Setup

```r
library(shiny)
#can define global stuff here
ui <- fluidPage(...) #this is an object we are creating an assigning to UI
server <- function(input, output) { ... }  #whereas this is an R function
shinyApp(ui = ui, server = server)
```

- ui is what the user can see and interact with
- server defines what gets displayed for the user to see, and what happens when the user interacts with the ui
- Example from `runExample("01_hello")` (`rshiny.gwdg.de/apps/sample-apps/hello-showcase/` or RStudio)



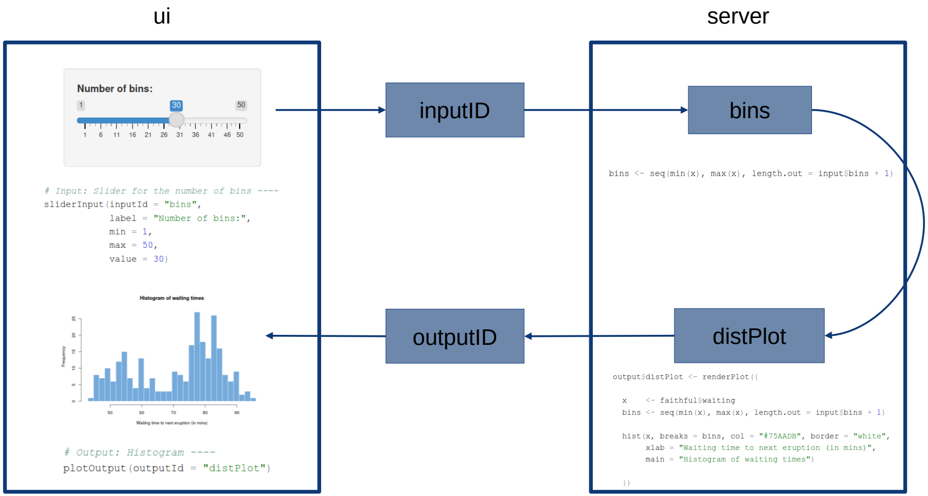
```r
library(shiny)

# Define UI for app that draws a histogram ----
ui <- fluidPage(

  # App title ----
  titlePanel("Hello Shiny!"),

  # Sidebar layout with input and output definitions ----
```

Diagram



```
# Input: Slider for the number of bins ----
sliderInput(inputId = "bins",
            label = "Number of bins:",
            min = 1,
            max = 50,
            value = 30)
```

```
# Output: Histogram ----
plotOutput(outputId = "distPlot")
```

```
bins <- seq(min(x), max(x), length.out = input$bins + 1)
```

```
output$distPlot <- renderPlot({

  x    <- faithful$waiting
  bins <- seq(min(x), max(x), length.out = input$bins + 1)

  hist(x, breaks = bins, col = "#75AADB", border = "white",
       xlab = "Waiting time to next eruption (in mins)",
       main = "Histogram of waiting times")

  })
```

ui

```r
ui <- fluidPage(
    titlePanel("Hello"),
    sidebarLayout(sidebarPanel(sliderInput(inputId = "", ...)),
                  mainPanel(plotOutput(outputId = "distPlot"))
                 )
)
```

- Create a "fluid page"
- with a given title
- and a two pane layout
- which we can fill with inputs and outputs, each with a named ID

server

```r
server <- function(input, output) {
    output$distPlot <- renderPlot({
        x    <- faithful$waiting #retrieve a dataset
        #retrieve values from slider
        bins <- seq(min(x), max(x), length.out = input$bins + 1)
        hist(x, breaks = bins)
    })
}
```

- We reference previous input and output IDs

Some special features

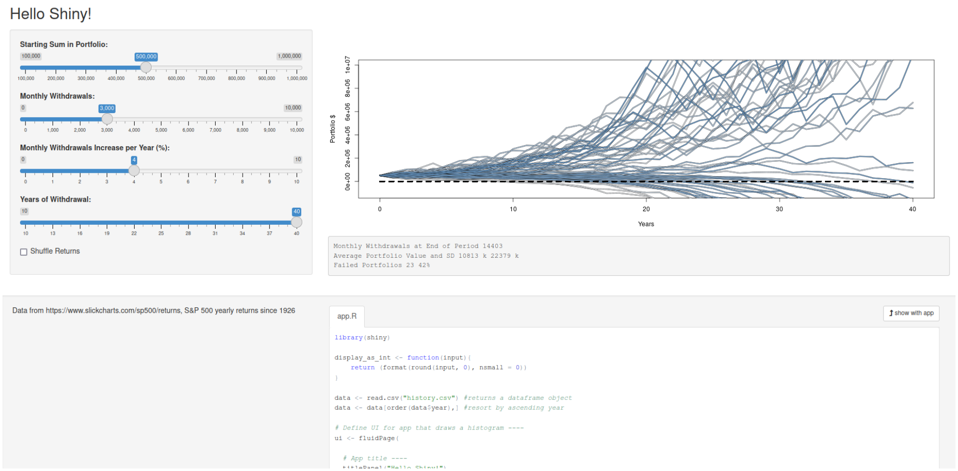
Can stack HTML objects:

```
mainPanel(
    h1("First level title"),
    h2("Second level title"),
    h3("Third level title"),
    h4("Fourth level title"),
    h5("Fifth level title"),
    h6("Sixth level title")
)
```

Images, need to be in a folder named www:

```
img(src = "my_image.png", height = 72, width = 72)
```

Example App


Hello Shiny!

Starting Sum in Portfolio: 500,000 (100,000 – 1,000,000)

Monthly Withdrawals: 3,000 (0 – 10,000)

Monthly Withdrawals Increase per Year (%): 4 (0 – 10)

Years of Withdrawal: 40 (10 – 40)

☐ Shuffle Returns

```
Monthly Withdrawals at End of Period 14403
Average Portfolio Value and SD 10813 k 22379 k
Failed Portfolios 23 42%
```

Data from https://www.slickcharts.com/sp500/returns, S&P 500 yearly returns since 1926

```
library(shiny)

display_as_int <- function(input){
    return (format(round(input, 0), nsmall = 0))
}

data <- read.csv("history.csv") #returns a dataframe object
data <- data[order(data$year),] #resort by ascending year

# Define UI for app that draws a histogram ----
ui <- fluidPage(

  # App title ----
  titlePanel("Hello Shiny!")
```

Figure: rshiny.gwdg.de/apps/sample-apps/portfolio_sim/, Inspired by firecalc.com

Going Forward

- RShiny website has very nice documentation and showcases
- Presentation on "advanced" features such as menu’s, downloading files, loading icons, changing tab header, hosting an RShiny Server, etc?
- Working on a front/backend presentation, accessing GPU resources at the SCC
- We already offer a server, we want more apps!
- Should we offer a test server, with limited resources and lower assured hosting time?
- Workshop to help people with their apps?